

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Приладобудівний факультет**

**Приладобудування**

До захисту допущено:

Завідувач кафедри

Юрій КИРИЧУК

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

зі спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології  
систем точної механіки»

на тему: «Система автоматичного зчитування показників лічильників»

Виконав (-ла):

студент (-ка) 4 курсу, групи ПМ-61

Кільчицький Андрій Орестович \_\_\_\_\_

Керівник:

Доцент, к.т.н.

Гришанова Ірина Аркадіївна \_\_\_\_\_

Консультант з технологічного розділу:

Професор, д.т.н.

Антонюк Віктор Степанович \_\_\_\_\_

Рецензент:

Доцент, к.т.н.

Шевченко Вадим Володимирович \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2020 року

# ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

[illegible]

				ДП ПМ61 06.000.00		
	ПІБ	Підп.	Дата			
Розробн.	Кільчиць кий			Відомість дипломного проєкту	Лист	Листі в
Керівн.	Гришано ва				1	1
Консуль т.	Антонюк				КПІ ім. Ігоря Сікорського Каф. ПБ Гр. ПМ-61	
Н/контр.						
Зав.каф.	Киричук					

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Факультет \_\_\_\_\_ Приладобудівний \_\_\_\_\_  
(повна назва)

Кафедра \_\_\_\_\_ Приладобудування \_\_\_\_\_  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність (спеціалізація) 151 Автоматизація та комп'ютерно-інтегровані технології (Комп'ютерно-інтегровані технології та системи точної механіки) \_\_\_\_\_  
(код і назва)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
\_\_\_\_\_ Юрій КИРИЧУК.  
(підпис) (ініціали, прізвище)  
«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
на дипломний проект студенту**

Кільчицькому Андрію Орестовичу

\_\_\_\_\_  
(прізвище, ім'я, по батькові)

Тема \_\_\_\_\_ Система автоматичного зчитування показників лічильників

керівник проекту Гришанова Ірина Аркадіївна, к.т.н., доцент

\_\_\_\_\_  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. № \_\_\_\_\_

2. Строк подання студентом проекту \_\_\_\_\_

3. Вихідні дані до проекту \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно

розробити)

Вступ. 1. Огляд і аналіз існуючих систем автоматичного зчитування показників лічильників (AMR). 2. Розробка схеми власної AMR-системи. 3. Підбір і розрахунок обладнання для створюваної системи. 4. Розробка алгоритму і програмного забезпечення для автоматичного зчитування показників лічильників. 5. Опис особливостей роботи системи. 6. Технологічний розділ. Висновки. Список використаних джерел.

5. Перелік (ілюстративного) графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів, тощо)

5.1 Схема системи автоматичного зчитування показників лічильників – 1арк.фА1

5.2. Алгоритм роботи системи – 1арк.фА1

5.3 Схеми, графіки, що пояснюють роботу системи – 2арк.фА1

#### 6. Консультанти розділів проекту\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Технологічний			

7. Дата видачі завдання \_\_\_\_\_

#### Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1.	Огляд і аналіз аналогів	12.03-01.04.2020	
2.	Розробка схеми системи автоматизованого зчитування	01.04-10.04.2020	
3.	Підбір обладнання і виконання розрахунків	11.04-21.04.2020	
4.	Розробка алгоритму збору даних	22.04-01.05.2020	
5.	Розробка опису роботи системи	02.05-12.05.2020	
6.	Виконання технологічного розділу	13.05-18.05.2020	
7.	Підготовка КД	01.05-22.05.2020	
8.	Підготовка ПЗ	18.05-22.05.2020	

Студент

\_\_\_\_\_ (підпис)

Кільчицький А.О.

\_\_\_\_\_ (ініціали, прізвище)

Керівник проекту

\_\_\_\_\_ (підпис)

Гришанова І.А.

\_\_\_\_\_ (ініціали, прізвище)

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Система автоматичного зчитування показників лічильників»**

Київ – 2020 року

### Анотація

Дипломний проект на тему «Система автоматичного зчитування показників лічильників» складається з 6 розділів, висновків, списку літератури і додатків. Пояснювальна записка містить 64 сторінки, 34 рисунків, 7 таблиць, список літератури з 10 найменувань та 1 додатка.

Тема дипломного проекту є актуальною, оскільки дана система дасть змогу заощадити не тільки витрати енергоресурсів і економічних ресурсів, а й найголовніше – людські сили та час. Використання даної системи значно спростить процес знімання показників лічильників витрат енергоресурсів, та зробить наочною статистику використання енергоресурсів користувачем. В сучасному світі людство всюди займається оптимізацією та мінімізацією витрат енергоресурсів, тому система, яка дає змогу не тільки оптимізувати процес зняття показників, а також зберігати і оброблювати дані про витрати за довгий період часу є важливою і актуальною.

Метою бакалаврського проекту є огляд AMR систем, принцип дії та архітектура таких систем, ознайомлення з аналогами та удосконалення даної системи.

Модулем удосконалення системи автоматичного зчитування показників лічильників є інформаційний ресурс, що виконує функції довгочасного збереження даних про витрати, їх аналізу та оптимізацію.

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		

## Annotation

Thesis project on "Automatic reading of meters" consists of 6 sections, conclusions, bibliography and appendices. The explanatory note contains 64 pages, 34 figures, 7 tables, a list of references of 10 titles and 1 appendix.

The topic of the diploma project is relevant, because this system will save not only energy and economic resources, but most importantly - human resources and time. The use of this system will greatly simplify the process of reading the indicators of energy consumption meters, and will make clear the statistics of energy use by the user. In today's world, humanity is everywhere optimizing and minimizing energy costs, so a system that allows not only to optimize the process of reading, but also to store and process data on costs over a long period of time is important and relevant.

The purpose of the bachelor's project is to review AMR systems, the principle of operation and architecture of such systems, acquaintance with analogues and improvement of this system.

The module for improving the system of automatic reading of meter readings is an information resource that performs the functions of long-term storage of cost data, their analysis and optimization.

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Пілпи	Лат		



## Зміст

Вступ.....	3
1. Огляд систем автоматичного зчитування показників лічильників	
1.1.Актуальність системи автоматичного зчитування показників лічильників .....	4
1.2.AMR.....	5
1.3.Основні принципи побудови САЗПЛ.....	6
1.4.Архітектура САЗПЛ.....	7
1.5.Варіанти збору та передачі показників.....	11
1.6.Аналоги САЗПЛ та їх виробники.....	16
1.7.Складові обраної САЗПЛ.....	16
1.8.WEB-ресурс.....	17
1.9.Висновки до розділу.....	26
2. Розробка схеми власної AMR.....	27
2.1.Схема AMR системи.....	27
2.2.Висновки до розділу.....	29
3. Підбір і розрахунок обладнання для створюваної системи.....	29
3.1.Підбір і розрахунок обладнання.....	29
3.2.Висновки до розділу.....	34
4. Розробка алгоритму і програмного забезпечення для автоматичного зчитування показників лічильників.....	35
4.1.Опис завдання удосконалення.....	35
4.2.Концептуальне проектування бази даних інформаційного ресурсу....	35
4.3.Даталогічна модель системи.....	36
4.4.Програми для проектування структури та контенту ресурсу.....	38
4.5.Структура інформаційного ресурсу.....	42
4.6.Висновок до розділу.....	43
5. Опис особливостей роботи системи.....	44
5.1.Опис особливостей роботи системи.....	44

5.2.Фізична структура інформаційного ресурсу.....	45
5.3.Функції для роботи з користувачем.....	46
5.4.Функції для роботи з базами даних.....	52
5.5.Функції для роботи з системою.....	53
5.6.Потенціал інформаційного ресурсу на розширення і вдосконалення..	54
5.7.Висновок до розділу.....	54
6. Технологічний розділ.....	55
6.1.Опис конструкції.....	55
6.2.Оцінка рівня технологічності.....	55
6.3.Розрахунок точності механоскладальних робіт.....	57
6.4.Координатний метод.....	58
6.5.Визначення фізичної (параметричної) взаємозамінності.....	59
6.6.Розробка технологічного процесу складання виробу.....	61
6.6.1. Схема складального складу.....	61
6.6.2. Технологічна схема складання.....	62
Висновки.....	64
Список літератури.....	66

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		

## Вступ

На сьогоднішній день створення інтелектуальних інформаційних систем, що забезпечують надання достовірних даних про надання житлово-комунальних послуг, є однією з найважливіших цілей в розвитку житлово-комунального господарства. Невід'ємною частиною таких систем є дистанційне одномоментне зняття показань приладів обліку витрат енергоресурсів.

На сьогоднішній день зростає чисельність населення, що призводить до скорочення числа доступних енергетичних і природних ресурсів. В цих умовах стає актуальним серйозний підхід до їх використання.

Автоматизовані системи обліку енергоресурсів - потужний інструмент в руках фахівців для отримання актуальної інформації про енергоспоживання об'єктів, її аналізу, виявлення та припинення безцільної витрати енергоресурсів.

Необхідною умовою інтеграції лічильників води, тепла, газу та електроенергії в сучасні системи диспетчеризації є взаємодія між самими лічильниками. Сьогодні вони повинні не тільки точно вимірювати, але також повинні бути легко включені в широкий спектр інтелектуальних вимірювальних систем через відповідні інтерфейси.

# **1. Огляд систем автоматичного зчитування показників лічильників**

## **1.1 Актуальність системи автоматичного зчитування показників лічильників**

На сьогоднішній день, одною із найактуальніших проблем все більше виражається проблема надлишкового використання енергоресурсів. Нераціональне використання енергоресурсів призводить не лише до економічних та практичних проблем, але і до екологічних. Надмірне використання енергоресурсів призводить до наступних негативних факторів:

- Забруднення навколишнього середовища через теоретичний надлишок добутих енергоресурсів;
- Переповнення фізичних сховищ енергоресурсів;
- Перенавантаження систем постачання;

Неабияк важливим також є недостовірний та неточний аналіз використання енергоресурсів. При неточному аналізі та обліку витрат енергоресурсів виникають недопустимі та незворотні втрати цих енергоресурсів. Перевтілення енергоресурсів в доволі дорогу одиницю, ставить перед світом якісно нові вимоги до вимірювання та обліку енергоресурсів.

Встановлення приладів обліку витрат енергоресурсів на об'єктах їх витрат є необхідною мірою та засобом покращення обліку та аналізу витрат в цілому. Але встановлення даних приладів не дає змогу зчитувати дані витрат та одночасно проводити їх глибокий аналіз. Так як прилади, встановлені на об'єктах витрат, чи то квартира, чи то приватний будинок, чи то велике підприємство, не мають зв'язку один з одним, можливий лише періодичний обхід об'єктів і збір даних, що в свою чергу, потребує невиправдані витрати зі сторони організацій в цілому.

Саме тому, реалізація системи, яка дасть змогу в повністю

автоматичному зборі та аналізі даних з приладів вимірювання витрат енергоресурсів є гостро актуальною. Дана системи дасть змогу заощадити не тільки витрати енергоресурсів і економічних ресурсів, а й найголовніше – людські сили та час.

## 1.2 AMR

AMR – це системи автоматичного зчитування показників лічильників. Система автоматичного зчитування показників лічильників призначена для автоматизованого дистанційного знімання, передачі, збору даних з різних приладів обліку (виміру) енергоресурсів споживачів комунально-побутового сектора (газ, вода, електроенергія, тепло).

Головною ідеологією таких систем є можливість заміни будь-якого з компонентів системи на аналогічний (іншого виробника), і це не буде мати ніякого впливу на загальну роботу системи.

Система зчитування даних по радіо каналу можуть складатися з наступних частин:

- лічильників тепла, що передають отримані дані по бездротовому каналу;

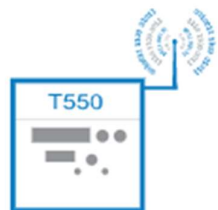


Рис. 1.1 Лічильник

- пристрою, що приймає дані з лічильників;



Рис. 1.2 Радіо-маршрутизатор

- комп'ютера, планшета або смартфона для відображення зібраних даних з лічильників;



Рис. 1.3 Пристрій відображення даних

- програмного забезпечення, встановленого на пристрої або сервері, з яким зв'язується девайс.



Рис. 1.4 Програмне забезпечення

Переваги інтелектуальних систем диспетчеризації:

- Ефективний процес отримання даних
- Миттєве створення звітів про витрати
- Майже стовідсоткова гарантія отримання даних
- Комфортний менеджмент даними
- Можливість відслідження споживання даних
- Прогнозування потенціалу економії
- Збереження ресурсів

### 1.3.Основні принципи побудови САЗПЛ

САЗПЛ використовує для збору даних бездротові системи зв'язку:

- для попереднього збору даних з приладів обліку енергоресурсів - радіоканали в не ліцензованому діапазоні частот;

- для відправки отриманих даних з метою зберігання та їх подальшого аналізу в інших автоматизованих системах - GSM (CDMA) канали зв'язку мереж зв'язку операторів стільникового зв'язку.

В якості головних приладів обліку енергоресурсів в САЗПЛ передбачається використання приладів обліку енергоресурсів з використанням імпульсних виходів. САЗПЛ має модульну структуру і будується на основі однотипних модулів. САЗПЛ повинна мати гнучку архітектуру для її адаптації до кардинально різних умов використання.

Встановлення та технічна експлуатація САЗПЛ має бути простою по своїй структурі та не повинна вимагати обширності знань від користувачів та обслуговуючого персоналу. Модулі САЗПЛ повинні мати низькі вимоги до живлення, що буде давати змогу використовувати модулі близько десяти років.

#### 1.4.Архітектура САЗПЛ

САЗПЛ має трирівневу архітектуру.

На *першому* рівні існують блоки збору та передачі даних, які виконують функції зв'язку з лічильниками витрат та передачі даних по радіоканалу до первинного пункту збору даних.

Лічильники витрат, призначені для вимірювання різних джерел ресурсів (газу, електроенергії, води та тепла) для комунального сектору.

У головній змінній застосуванні САЗПЛ повинен забезпечувати роботу з імпульсними джерелами живлення (розетка з відкритим колектором).



Рис. 1.5 Лічильник НІК 2104

Блок збору та передачі даних є повноцінним пристроєм і зазвичай встановлюється безпосередньо на лічильнику потужності. Його можна підключити до лічильника електроенергії безпосередньо, або за допомогою датчика імпульсу. Дані передаються з лічильників енергії через одиниці збору даних та заздалегідь надсилаються до бази даних. При використанні імпульсного виходу вимірювача потужності він підключається до блоку збору даних і передається багатожильним проводом. Бездротова антена блоку збору та передачі даних знаходиться безпосередньо в блоці, або може бути реалізована на певній відстані від неї.

Пристрій живиться від внутрішнього джерела живлення (акумулятора), розташованого всередині блоку. Ємність батареї пристрою повинна бути достатньою для забезпечення роботи пристрою для запису та передачі даних до 12 років (залежно від конкретного режиму використання, особливо часу передачі даних). Блок збору та передачі даних реалізований на базі одного з галузевих радіоагрегатів, що підтримують wM-Bus в мережах передачі даних по радіоканалу.





Рис. 1.6 Герконовий датчик імпульсів

На *другому* рівні є вузли первинного збору даних, які виконують функції вимірювальних приладів обстеження та збору даних по радіоканалу, а також залежно від типів вузлів, автоматизовану передачу даних до центру зберігання та обробки даних. Канали зв'язку GSM (CDMA) та системи зв'язку стільникового оператора використовуються для автоматичної передачі даних з об'єкта збору даних до центру зберігання та обробки даних.

Хаб-центри виконують роль фіксованих вузлів перед збором даних для отримання даних від блоків збору та передачі даних і від ретрансляторів.

Кількість та розташування затворних конденсаторів по відношенню до лічильників статичної енергії визначається просторовим розташуванням лічильників енергії та технічними можливостями концентраторів затвора в їх обстеженнях протягом певного періоду часу, що регулюється технічними вимогами.

Шлюз-концентратор зчитує дані з блоків збору та передачі даних, ретрансляторів, по радіоканалах, потім використовуючи стандартний модем GSM (або CDMA), який передає дані, отримані по стільниковій мережі, до центру зберігання та обробки даних.

В ситуації, якщо окремі одиниці для захоплення та передачі даних, які потрапляють у діапазон дії шлюзів, з однієї технічної причини не можуть бути допитані протягом організованого періоду часу, з метою збільшення

діапазону передачі даних та сили сигналу одиниць збору та передачі даних можуть бути використані ретранслятори.

Інформацію про його конфігурацію можна отримати віддалено через портальні концентратори блоку збору та передачі даних.

Передача даних від концентратора до центру зберігання та обробки даних може здійснюватися з ініціативи центру здійснюватися по ініціативі обох сторін в залежності від режимів роботи мережі та поточного стану системи. Час, розмір та швидкість передачі даних створюються при створенні концентратора.

Портал дистрибутора може містити кілька радіоантен для роботи на радіоканалі, які можна встановлені на певній відстані від них (до 7 метрів).

Джерело живлення концентратора відбувається від зовнішнього джерела живлення. Щоб отримати стабільне джерело живлення, розподільний затвор повинен використовуватися разом із гарантованим джерелом живлення, який повинен забезпечити його працездатність у разі тимчасового відключення електроенергії.

Оскільки зовнішні антени можуть бути встановлені, затвор конденсатора повинен мати громовідвід, щоб запобігти пошкодженню шляхів прийому пристрою під час грози. Хаб-шлюз можна налаштувати віддалено через канал зв'язку GSM (CDMA) або локально за допомогою персонального комп'ютера.

Хаб реалізований на базі одного із стандартних для промисловості радіоагрегатів, що підтримують wM-Bus в бездротових мережах передачі даних та стандартного модему GSM (CDMA).

На *третьому* рівні знаходиться центр накопичення та зберігання даних. Дані надсилаються з цього центру для подальшої обробки інших робототехнічних систем.

Центр збору та зберігання даних - це веб-ресурс, розміщений на FTP-сервері.

## 1.5 Варіанти збору та передачі показників

### *Стационарний спосіб збору та передачі показань приладів обліку*

При "стационарному" використанні, дані отримані за допомогою лічильників витрат енергоресурсів надсилаються частини системи, що відповідає за попереднє зберігання даних та подальшої їх відправки до частини обліку та зберігання даних.

Роль стационарних вузлів попереднього збору даних грають в цій ситуації виконують стационарні прилади попереднього збору даних – концентратори, які розташовуються в місці з найнадійнішим зв'язком с приладами обліку та вимірювання витрат енергоресурсів. У разі нечіткого зв'язку встановлюються ретранслятори, що виконують функцію повторення сигналу з приладів вимірювання енергоресурсів.

Число і місце розташування концентраторів по відношенню до приладів вимірювання витрат енергоресурсів, визначається властивостями самих концентраторів, їх потужності і дальності сигналу, властивостями конструкції об'єкту розташування системи автоматичного зчитування показників лічильників.

Отримані концентратором дані з приладів вимірювання витрат енергоресурсів обробляються вбудованим програмним забезпеченням, та надсилаються на FTP сервер, на якому в подальшому відбувається обробка отриманих даних.

Для збільшення дальності передачі даних з модулів знімання і передачі даних на вузли попереднього збору даних і розширення зони охоплення приладів обліку енергоресурсів з модулями знімання і передачі даних, в

системах автоматичного зчитування показників лічильників можуть бути використані спеціальні пристрої - ретранслятори.



Рис. 1.7 Стационарна система Z.RTU

Стационарна система Z.RTU - складається з пристрою реєстрації даних(RTU1), wireless ретрансляторів (Z.RPT) для збільшення зв'язку між пристроями в середині об'єкта.

Система має сумісність з розширеним спектром приладів обліку води та тепла, оснащених бездротовим інтерфейсом зв'язку wM-Bus (протокол 868 МГц, режим C1, T1, T2 або S1). Завдяки такій програмі, дані можуть збиратися та зберігатися протягом десяти років.

Функції інтелектуального обліку:

- Реєстратор даних на вбудованому веб-сервері
- Можливість одночасного керування до 500 приладами
- 16 ретрансляторів, що використовуються для розширення мережі
- Журнал зчитаних даних зберігаючий дані до 10ти років
- Зберігання даних з інтервалами від 15 хвилин до 1 місяця
- Можливість дистанційного керування системою, формування звітів і подальшої їх відправки на сервер або на електронну пошту.
- Дистанційні та автоматичні оновлення вбудованого програмного забезпечення.

Конструктивні особливості:

- Ступінь: захисту IP20;

- Кріплення: 35 мм DIN-рейка
- Розміри: 4 DIN-модуля (90х72х64.5)

Стационарна система збору та передачі даних «Z.RTU» призначена для:

- Лічильників холодної і гарячої води, обладнаних EDC-модулем
- Лічильників холодної і гарячої води, підключених до PDC-модулю
- Теплолічильників «Цельсіус», оснащених інтерфейсом зв'язку wM-Bus



Рис. 1.8 Схема збору та передачі показань приладів стаціонарним способом

Переваги стаціонарного способу збору і передачі даних:

- Автоматичний збір і передача даних без участі персоналу
- Відсутність витрат на персонал для збору показань приладів обліку

### ***Мобільний спосіб збору та передачі показань приладів обліку Walk-by***

При використанні мобільного способу збору показників лічильників використовуються мобільні прилади збору та передачі даних, які в свою чергу проводять опитування блоки попереднього збору даних з приладів вимірювання витрат на об'єкті.

Роль групування даних з приладів вимірювання витрат мають комунікатори з підключеним до них блоком прийому даних по радіоканалу.

Оператор з мобільним приладом збору і передачі даних обходить об'єкт в якому встановлені прилади обліку витрат енергоресурсів. При цьому немає необхідності заходити в середину об'єкту, так як передача даних відбувається дистанційно по радіоканалу.

Після того як оператором було проведено зчитування даних з об'єктів, він переносить отримані дані в центр накопичення і збереження даних.

Передача даних в даному способі проходить з використанням стандартних засобів, протоколів та інтерфейсів обміну інформацією, які є стандартами в інформаційних системах.



Рис. 1.9 пристрій збору даних «MinoConnect»

Пристрій збору даних «MinoConnect» призначений для дистанційного збору даних по радіоканалу з лічильників, оснащених інтерфейсом зв'язку wM-Bus (868 МГц).

Конструктивні особливості:

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		
						14

- Вбудований акумулятор
- Індикація рівня заряду батареї
- Індикація режиму роботи
- З'єднання з планшетом або телефоном на базі Android через інтерфейс Bluetooth.
- Інтерфейс зв'язку RS232 і RS485

Пристрій збору даних «MinoConnect» призначене для:

- Лічильників холодної і гарячої води, обладнаних EDC-модулем
- Лічильників холодної і гарячої води, підключених до PDC-модулю
- Теплолічильників «Цельсіус», оснащених інтерфейсом зв'язку wM-Bus

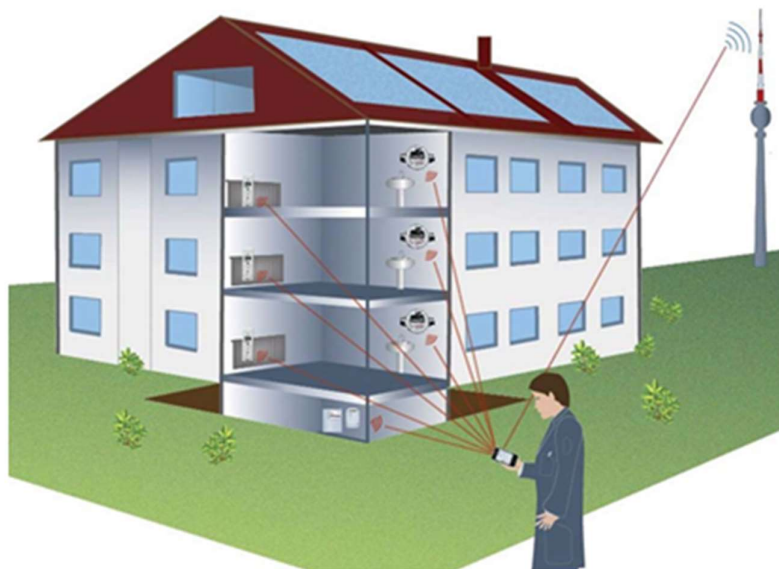


Рис. 1.10 Схема збору та передачі показань приладів мобільним способом

Переваги мобільного способу збору і передачі даних:

- Автоматизований збір і передача даних;
- Гарантований збір свідчень;
- Істотне скорочення первинних витрат на оснащення будівлі обладнанням
- Істотне скорочення витрат на обслуговування обладнання

## 1.6 Аналоги САЗПЛ та їх виробники

L&G Meetering

Виробляють САЗПЛ за використанням радіотехнологій (збір даних з лічильників тепла по бездротовому радіо каналу на частоті 868 МГц (wireless M-Bus)) та GSM / GPRS (автоматизовані системи збору та передачі даних для квартирних лічильників тепла).

ZENNER

Системна технологія ZENNER забезпечує модульні структуровані рішення, адаптовані до індивідуальних вимог клієнтів. Компанія виробляє як провідні шини M-Bus, так і бездротові радіорешення wM-Bus, а також інтелектуальні рішення для інтеграції в вимірювальні системи звичайних лічильників з імпульсним виходом.

ISTA International GmbH

Система радіосбору даних «Сімфонік»,

Система автоматичного збору даних по радіоканалу мод. «Symphonic Sensor net» (далі: «Сімфонік») виробництва компанії «ISTA International GmbH» (торгова марка: «ISTA, ISTA-RUS», далі ICTA, ICTA-РУС) призначена для дистанційного збору даних з приладів обліку різних типів і виробників без доступу в приміщення і на об'єкти з метою здійснення комерційного обліку споживання комунальних ресурсів.

## 1.7.Складові обраної САЗПЛ

Система збору даних складається з трьох частин:

1. Клієнтської частини (радіомодулі, прилади обліку, концентратори, інтернет пристрій (комп'ютер з доступом в інтернет і т.п.)) зі стаціонарним способом зняття .
2. Канали передачі даних: стільниковий зв'язок стандарту GSM та Інтернет.
3. «Інформаційного ресурсу» (WEB-ресурс) -для дистанційного скачування звітів за обраний період, огляду даних с лічильників у реальному часі, розрахування тарифів та т.п.



Особливості:

Обраний спосіб отримання показаннями приладів обліку, а саме: свідчення зібрані концентраторами спочатку передаються на веб-портал з об'єкту через GSM, а потім через мережу інтернет користувач самостійно завантажує звіт з «Інформаційного ресурсу», надає користувачу системи вийняткові зручності:

- Результати доступні на «Інформаційному ресурсі» через інтернет в будь-який час, в будь-якому місці, з будь-яким смартфоном, планшетом, ноутбуком або комп'ютером (тобто на будь-якій операційній системі і платформі без обмеження);

- на «Інформаційному ресурсі» користувач системи, самостійно та автоматично може скачати звіти, які створюються ресурсом;

- «Інформаційний ресурс» може створювати звіти за минулі розрахункові періоди, так як є зовнішнім тимчасовим сховищем свідчень за досить великий період часу (аналогічно «хмарним інтернет технологіям»).

### **1.8. WEB-ресурс[1,2,3,4,12,13]**

WEB-ресурс розроблено на мовах програмування PHP, JavaScript. Загальна архітектура ресурсу базується на основі патерну розробки MVC (Model-View-Controller) за допомогою фреймворку Laravel. База даних розроблена на мові програмування SQL.

Так як веб ресурс необхідно розташувати на FTP сервері для того щоб система мала можливість відправляти дані, а користувачі могли користуватись цим ресурсом, розглянемо FTP сервер, його призначення та особливості.

### **FTP сервер**

FTP-сервер – сервер, працюючий по принципу File Transfer Protocol (протоколу передачі файлів). Використовується для обміну файлами між комп'ютерами по локальній мережі і інтернету.

Дана веб технологія є одною із самих затребуваних для скачування і загрузки даних з/на віддалений сервери.

FTP-системи широко поширені в розробці веб сайтів. Вся необхідна інформація, що відноситься до сайту знаходить на сервері. Тобто при роботі з сайтом, який знаходиться на FTP-сервері, користувач напряду зв'язується з цим сервером.

Якщо розробнику сайту необхідно внести будь-які зміни на FTP сервері, існує дуже багато спеціалізованих програм для роботи з FTP серверами.

Для входу на сервер необхідно пройти процедуру авторизації. Існують різні способи авторизації на FTP-сервер:

- Авторизація за допомогою логіну і пароля
- Анонімний доступ (система автоматично зашифровує дані про користувача і надає обмежені права, загалом використовується для оновлення ПО)

Приклади програм для роботи з FTP-серверами:

- Total Commander
- FileZilla
- Будь-який сучасний браузер

Також існує програмне забезпечення для розробників програмного забезпечення, яке надає можливості загрузки даних на FTP-сервер:

- GIT
- GITLab

Розглянемо обрані середовища та обґрунтування їх вибору:

## **RНР**

Абревіатурою RНР називають один з найбільш популярних мов, що використовуються-ваних для створення веб-додатків. Спочатку RНР означало Personal Home Page - персональна домашня сторінка, але потім, з розширенням

області при-трансформаційних змін мови, аббревіатура PHP стала розшифровуватись PHP: Hypertext Preprocessor - PHP: гіпертекстовий препроцесор. Поточна версія мови PHP призначена для створення веб-додатків найширшого призначення, напри-заходів, на його основі створена відома соціальна мережа Facebook. В даний час PHP підтримується переважною більшістю хостинг-провайдерів, його використовують сотні тисяч програмістів[1].

Особливості:

Особливістю даної мови програмування є те, що в файлі PHP можна використовувати не лише код мови PHP, але й мови HTML та JavaScript. Це дає можливість ефективно оперувати мовами для розробки веб застосунків і розробляти найрізноманітніші архітектури застосунків.

PHP має велику бібліотеку вбудованих функцій, що дозволяє полегшувати рутинну розробку і уникати багаторядкового опису функцій.

#### *Наявність інтерфейсів до багатьох баз даних*

- у PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, SQLite, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase.
- завдяки стандарту відкритого інтерфейсу зв'язку з базами даних (англ. Open Database Connectivity Standard, ODBC) можна підключатися до всіх баз даних.

#### *Нетрадиційність*

Мова PHP являється досить знайомою для розробників, що працюють в різних областях програмування. Мова PHP має багато сходжень з мовою C та іншими відомими мовами програмування. Це значно знижує необхідний час при вивченні PHP. PHP поєднує переваги Perl та C і спеціалізована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. Не дивлячись на те, що PHP є досить молодого мовою, вона здобула неабияку популярність

серед web-програмістів. В наш час PHP є найпопулярнішою мовою для створення веб-застосунків (скриптів). На мові PHP базуються більше 80% всіх веб ресурсів в інтернеті.

### *Ефективність*

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. За деякими оцінками, більшість PHP-сценаріїв обробляються швидше за аналогічні їм програми на інших мовах програмування. Не дивлячись ні на що, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше — в десятки, а іноді і в сотні разів, ніж інтерпретовані.

Саме тому оптимізація коду в PHP є дуже важливим аспектом розробки застосунків. Адже найбільше швидкості втрачається на неякісних і не оптимізованих сценаріях.

### **Laravel**

Laravel — безкоштовний PHP-фреймворк, головною перевагою якого є відкритий код. Призначений для розробки веб-додатків відповідно до шаблону model–view–controller (MVC). Особливостями Laravel є:

- модульна архітектура з виділеним менеджером залежностей Composer;
- широка варіація способів доступу до реляційних баз даних;
- утиліти, які допомагають в розгортанні додатків і технічного обслуговування;
- орієнтація на синтаксичний цукор.

Сирцевий код Laravel'a розміщується на GitHub і ліцензований відповідно до умов MIT License.

## MVC

Модель–вид–контролер — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

MVC складається з об'єктів трьох видів. Модель - це об'єкт докладання, а вид - екранне уявлення. Контролер описує, як інтерфейс реагує на дії користувача. До появи схеми MVC ці об'єкти в призначених для користувача інтерфейсів змішувалися. MVC відокремлює їх один від одного, за рахунок чого підвищується гнучкість і поліпшуються можливості повторного використання[2].

На рисунку нижче зображені модель і три вигляди. Модель містить дані, які можуть бути представлені у вигляді електронної таблиці.

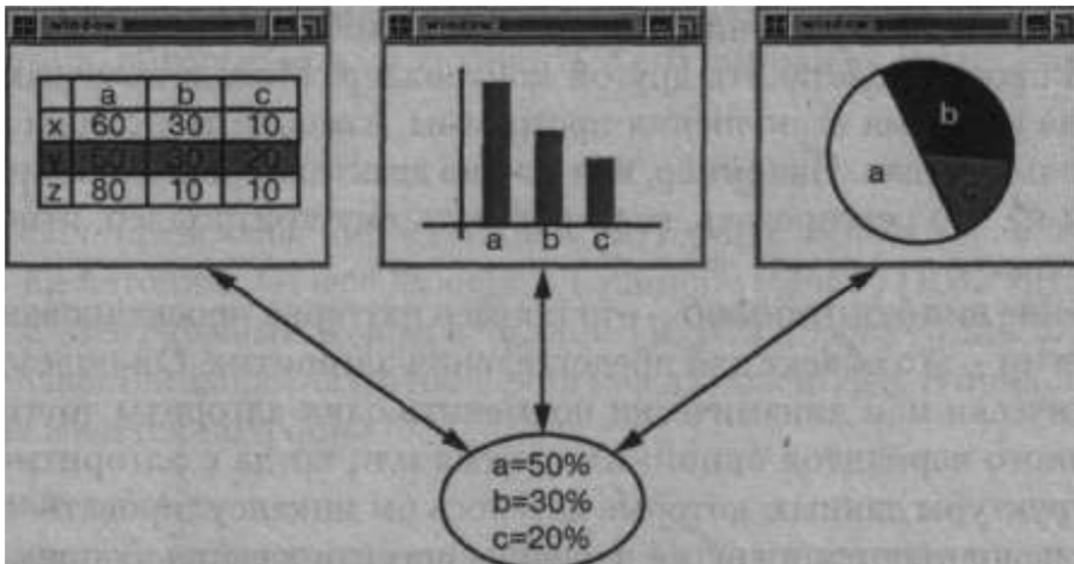


Рис. 1.11 – Модель-вигляд

Модель сповіщує свої види при кожному внесенні змін в дані, а вигляди звертаються до моделі за оновленням даних.

Ще одна властивість MVC полягає в тому, що види можуть бути вкладеними. Наприклад, панель управління, що складається з кнопок, допустимо представити як складовий вид, що містить вкладені, - по одній кнопці на кожен. Інтерфейс інспектора об'єктів може складатися з вкладених видів, які використовуються також і в отладчике. MVC підтримує вкладені види за допомогою класу CompositeView, що є підкласом View. Об'єкти класу

CompositeView поведуться так само, як об'єкти класу View, тому можуть використовуватися всюди, де і види. Але ще вони можуть містити вкладені види і управляти ними[2].

#### Опис та функції компонентів

В рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма ділиться на три частини, розподіляючи функції між компонентами. Модель (Model) відповідальна за збереження даних та їх структуру. Вигляд (View) відповідає за відображення цих даних користувачеві, тобто інтерфейс користувача. Контролер (Controller) відповідає за керування компонентами, отримання сигналів у вигляді реакції на дії користувача і передачу даних у модель.

- Модель – центральний компонент шаблону MVC, який відображає поведінку програми, незалежно від користувальницького інтерфейсу. Модель стосується правил прямого управління даними, логікою та правилами застосування..
- Вигляд може приймати форму будь-якого подання інформації, отриманої на виході, наприклад таблиці або графіка. Одночасно можуть співіснувати декілька виглядів (уявлень) однієї інформації, такі як графік управління компанією та графіки роботи аналітиків.
- Контролер отримує вхідні дані та перетворює їх у логічні команди для моделі або вигляду.

Модель займається інкапсуляцією ядра даних і основним функціоналом їхньої обробки і не залежить від процесів вводу чи виводу даних.

Задачами контролера є прослуховування будь-яких подій, що виникають в результаті дій користувача. За допомогою групування логічно пов'язаних дій в окремий клас, реалізується структуризація коду.

Задля незалежного використання компонентів екранізації даних, використовується відокремлення моделі від представлення. Таким чином, інформація буде автоматично редагуватися в залежності від внесених користувачем змін в той чи інший компонент системи.

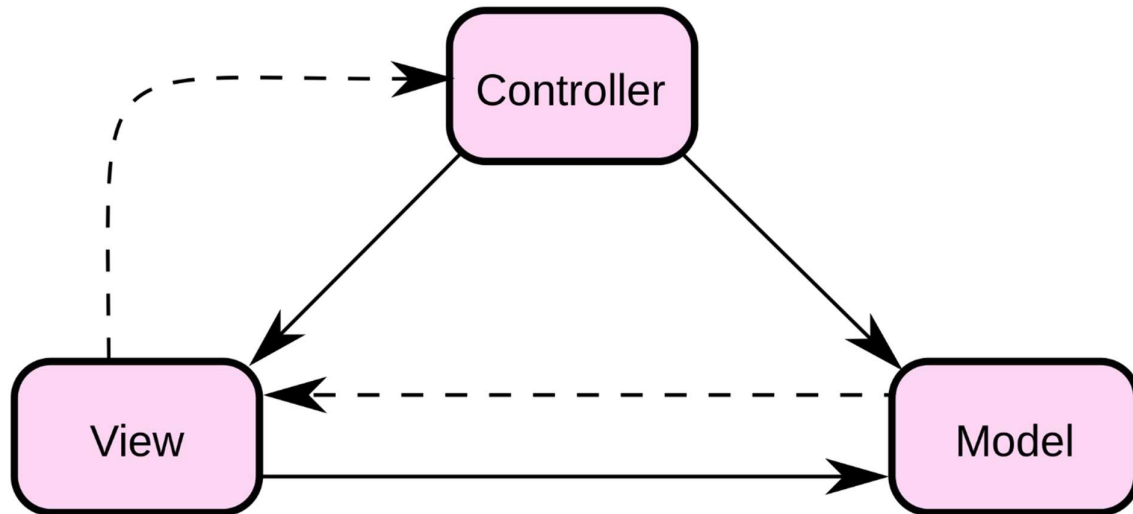


Рис. 1.12 Структура шаблону MVC

#### JavaScript

JavaScript ("JS" для стислості) - це повноцінний динамічний мову програмування, який застосовується до HTML документу, і може забезпечити динамічну інтерактивність на веб-сайтах. Його розробив Brendan Eich, співзасновник проекту Mozilla, Mozilla Foundation і Mozilla Corporation[3].

JS являє собою дуже універсальну, компактну та гнучку структуру. Розробники написали велику кількість інструментів для даної мови програмування, що даються змогу розширити можливості використання JavaScript.

JavaScript класифікується як прототипна, скриптова мову програмування з динамічною типізацією. JavaScript підтримує і інші парадигми програмування (імперативну та частково функціональну), деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація,

автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання скриптів веб-сторінок для надання їм динамічності;
- створення веб-застосунків з одною сторінкою (React, AngularJS, Vue.js);
- програмування серверної частини сайту (Node.js);
- стаціонарні застосунки (Electron, NW.js);
- застосунки на смартфони (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- робота з PDF-документами.

Мови Java та JavaScript є двома різними мовами, не дивлячись на схожі назви. Вони мають різну семантику, не дивлячись на те що й мають спільні риси в стандартних бібліотеках та правилах іменування. Синтаксис даних мов також отриманий «у спадок» від мови C.

JS має деякі властивості об'єктно-орієнтованої мови, але за допомогою концепції прототипів підтримка об'єктів в ній значно відрізняється від традиційних мов з використанням ООП.

## SQL

База даних – це фактично сукупність взаємопов'язаних даних.

Наприклад бібліотека, це база даних книг, які мають різний жанр, різних авторів, різні роки видання і тд.

Без сумніву, будучи широко використовуваною, бібліотека не ідеальна база даних і має деякі недоліки:

- Пошук потрібної книги, може займати велику кількість часу, якщо в бібліотеці знаходиться велика кількість книг.
- Бібліотека потребує постійного пересортування книг, що займає також велику кількість часу



- Індексція книг в бібліотеках завжди обмежена, тому пошук необхідних книг по деталям самої книги завжди буде обмежений

Реляційна база даних – база даних, в якій дані представлені у вигляді таблиць, взаємопов’язаних між собою.

SQL - Мова програмування для взаємодії користувачів з базами даних, що використовується для створення запитів, оновлення та управління реляційними базами даних, створення та зміни схеми бази даних та системи контролю доступу до бази даних. Сам SQL не є ні системою управління базами даних, ні окремим програмним продуктом.

SQL — мова програмування діалогу для запитів та внесення змін до бази даних, а також управління базою даних.

Переваги:

*Незалежність від конкретної СУБД*

Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку орієнтувалися на застосування щонайменше кількох СУБД. Природно, що при застосуванні деяких специфічних для реалізації можливостей такої переносимості добитися вже дуже важко[4].

*Наявність стандартів*

Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах[4].

*Декларативність*

За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо

при обробці SQL-запиту. Однак не варто думати, що це повністю універсальний принцип - програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту. Чим складніше сконструйований запит, тим більше він допускає варіантів написання, різних за швидкістю виконання, але однакових за підсумковим набору даних[4].

### **1.9 Висновки до розділу**

В даному розділі було розглянуто актуальність систем автоматичного зчитування показників лічильників, особливості їх архітектури та будови. Опираючись на існуючі галузеві проблеми, недоліки напівавтоматизованих систем зчитування показників лічильників, що базуються безпосередньому на фізичному контакті з лічильниками, вирішена проблема актуальності системи та визначені головні переваги, що може надати впровадження даної системи в об'єкти витрат.

Особливості структури та архітектури надають можливості її гнучкого налаштування та обслуговування, так як система має модульну структуру. Опираючись на модульну структуру, можна прийти висновку, що заміна одного з модулів системи, не буде впливати на роботу системи в цілому. Це є одною із найважливіших переваг даної системи.

Було розглянуто особливості побудови інформаційних ресурсів для даної системи, та інструменти, що використовуються для розробки і підтримки інформаційних ресурсів.

## **2. Розробка схеми власної AMR**

### **2.1. Схема AMR системи**

Проаналізувавши існуючі AMR системи, їх структури та принципи роботи було обрано стаціонарну бездротову AMR систему для подальшого удосконалення.

Бездротові технології дають можливості для встановлення даної систему на будь-якому об'єкті.

На рисунку 2.1 зображена схема власної AMR системи.

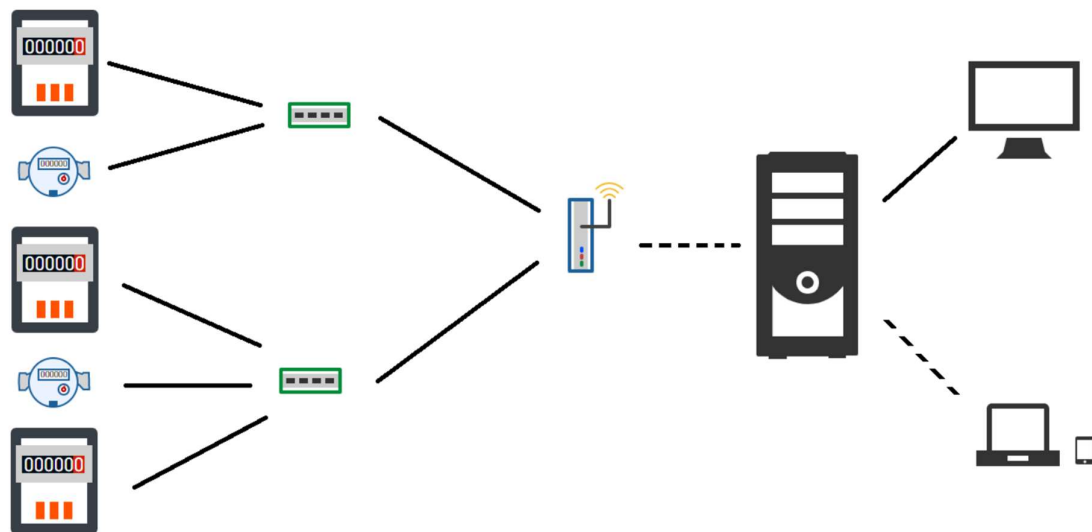


Рис. 2.1 – схема AMR системи.

Проаналізуємо розроблену схему:

На першому рівні схеми розташовані вимірювальні прилади (лічильники). Вони обладнані радіонакладками за допомогою яких відбувається передача даних на наступний рівень системи.

На другому рівні розташовані ретранслятори. Ретранслятори призначені для дублювання і збільшення дистанції передачі даних від вимірювальних приладів до концентратора. Використання ретрансляторів не завжди є раціональним і доцільним, адже якість бездротового зв'язку залежить від матеріалу стін, їх товщини та загальної площі об'єкту.

На третьому рівні розташований концентратор. Прилад, що проводить зчитування показників з приладів обліку енергоресурсів. Програмна частина приладу дає змогу формувати CSV файли з даними витрат вимірювальних приладів та відправляти їх на вказаний FTP сервер.

Четвертий рівень системи – FTP сервер. На ньому зберігаються всі дані про витрати лічильників користувачів, працює веб ресурс, що оброблює дані користувачів та відображає їх з необхідними фільтрами.

На п'ятому рівні розташовані користувачі. Вони мають змогу отримати реальні дані витрат їх лічильників з будь-якого пристрою з браузером. За вказаною інтернет-адресою вони з'єднуються з веб сервером через інтерфейс користувача.

Для комунікації між собою, прилади на об'єкті використовують протокол бездротової передачі даних wM-Bus.

wM-Bus (wireless Meter-bus) – визначає лінії частот та способи взаємодії між пристроями збору даних, лічильниками води, газу та електроенергії. Тобто є стандартом бездротового зв'язку між пристроями в конкретних частотах.

На даний момент за даним стандартом закріплені наступні режими роботи:

- N-mode – варіант стандарту для полоси в 169 МГц.
- F – варіант стандарту для полоси 433 МГц.
- S,T та C застосовуються в полосі частот 868 МГц.

## 2.2. Висновки до розділу

Опираючись на проаналізовані аналоги систем автоматичного зчитування було розроблено власну схему системи AMR. Система базується на протоколі бездротової передачі даних wM-Bus, має гнучку структуру та

достатній рівень розширення системи, що дозволяє встановлювати її на різноманітних об'єктах.

### **3. Підбір і розрахунок обладнання для створюваної системи**

#### **3.1 Підбір і розрахунок обладнання**

Підбір обладнання для AMR системи є важливим етапом її розробки, адже від вибору оптимального обладнання залежить складність встановлювання системи, ціна її встановлення та загальна доцільність встановлення даної системи.

Так як система надає можливості бездротової передачі та обліку витрат лічильників води, газу та електроенергії, необхідно врахувати особливості лічильників даних енергоресурсів.

Для передачі показників витрат води, використовуються універсальні накладки для комунікації. Вони призначені для бездротової передачі вимірювальних даних з протоколом wM-Bus.

Принцип дії накладки заснований на скануванні спеціальної стрілки лічильного механізму водного лічильника за допомогою індукційної системи, яка виявляє і розпізнає напрямки її обертання.

APT-WMBUS-NA-1

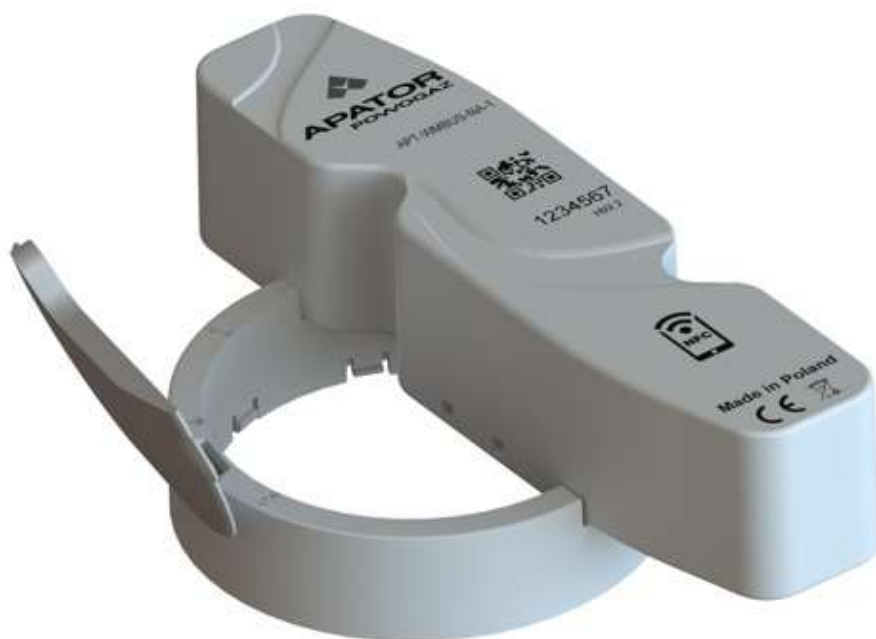


Рис. 3.1 – Радіонакладка APT-WMBUS-NA-1

Особливості даної накладки:

- Модуль має широкий спектр сумісності з квартирними, домовими та промисловими лічильниками.
- При монтажі даного модуля немає необхідності поглиблення в сам лічильник
- Висока стійкість до зовнішнього магнітного поля
- Розпізнавання напрямку потоку
- Клас захисту IP68

Технічні характеристики:

Таблиця 3.1 – технічні характеристики APT-WMBUS-NA-1

Радіус дії на відкритій місцевості	< 800 м
Протокол	wM-Bus*
Частота передачі даних	868,95 МГц
Вихідна потужність передавача	20 мВт / 50 Ом

Клас захисту	IP68
Маса	0,106 кг

Додавання лічильників в AMR можливе також за допомогою перетворювача. Необхідна наявність імпульсного виходу на лічильниках.

У разі якщо лічильники мають імпульсні виходи, в систему додається перетворювач імпульсів в протокол wM-Bus.

Перетворювач імпульсів PadPuls M2



Рис. 3.2 – перетворювач імпульсів PadPuls M2.

Призначений для підключення вимірювачів витрат таких як електrolічильники, лічильники газу, води до системи wM-Bus.

Ретранслятор AT-WMBUS-05



Рис. 3.3 – ретранслятор AT-WMBUS-05.

Ретранслятор є пристроєм-посередником між радіонакладками (модулями) та концентратором. Ретранслятор призводить до збільшення максимально допустимої відстані між даними пристроями. Ретранслятор працює по принципу повторної передачі прийнятих даних з радіомодулів.

Технічні характеристики:

Таблиця 3.2 – технічні характеристики AT-WMBUS-05

Живлення	~ 230 V, гальванічна ізоляція за допомогою трансформатора або від батареї
Споживання потужності	<1W
Робоча температура	от 0 °C до 55 °C
Клас захисту	IP54
Маса	0,172 кг



## Концентратор даних EVM-03



Рис. 3.4 – концентратор даних EVM-03

Технічні характеристики:

Таблиця 3.3 – технічні характеристики EVM-03

Цифрові інтерфейси (назва / кількість)	Оптичний - 1; RS-232 - 3; RS-485 - 1.
Оптичний порт	За рекомендацією ІЕС 1107
Швидкість обміну через оптичний інтерфейс, біт / с	9600
Напруга живлення, В	12 ... 24
Наявність архівів	Архів коригувань

Діапазон робочих частот GSM, МГц	850/900/1800/1900
Клас GPRS	12
Клас GSM	4
Споживана потужність (не більше), ВА	6
Клас захисту від ураження електричним струмом по ГОСТ 12.2.091-2002	ІІІ
Ступінь захисту оболонки по ГОСТ14254-96	IP52
Діапазон робочих температур	0 ° С ... +55 ° С
Середній термін експлуатації (не менше), років	12
Середнє напруцювання (не менше), годин	50000
Габаритні розміри, мм	192 x 137 x 49
Маса (не більше), кг	1,0

### 3.2. Висновки до розділу

Проаналізувавши існуючі прилади для систем автоматичного зчитування показників лічильників, їх аналоги та принципи роботи, було обрано оптимальні складові системи. Так як систему спроектовано на протоколі бездротової передачі даних wM-Bus, було обрано підтримуючі дану технологію прилади. Обране обладнання повністю відповідає поставленій задачі, обладнання легке у встановленні та обслуговуванні.

## 4. Розробка алгоритму і програмного забезпечення для автоматичного зчитування показників лічильників

### 4.1 Опис завдання удосконалення

Розглянувши в попередніх розділах структуру систем автоматичного зчитування показників лічильників, визначивши оптимальні для використання складові системи, було прийняте рішення удосконалити інформаційний ресурс.

*Зміст завдання:*

Удосконалення контролю витрат лічильників на об'єкті, користувачем, за допомогою інформаційного ресурсу з даними про витрати які оновлюються у реальному часі.

*Напрямки застосування:*

Будь які об'єкти на яких використовуються лічильники на газ, воду та електрику (від приватного дому або квартири до промислових споруд).

*Вигоди для користувача:*

- Зручний інтерфейс для користувача надає всі необхідні функції для контролю та обліку витрат лічильників
- Можливість перевірки та передачі даних на будь-якому пристрої з браузером
- Прискорення передачі даних лічильників та відсутність безпосереднього контакту для зняття даних.

### 4.2 Концептуальне проектування бази даних інформаційного ресурсу

Концептуальне проектування - процес створення моделі використовуваної на підприємстві інформації, що не залежить від будь-яких фізичних аспектів її представлення. Такий процес проектування створюється не опираючись на будь-яку конкретну систему управління базами даних.

ER-модель або ER-діаграма – це модель даних, яка надає можливість опису концептуальних схем предметної області. Моделлю даних є графічний

опис предметної області за допомогою стандартизованого набору позначень.

Зв'язок в базі даних – графічно відображена асоціація, встановлена між двома різними сутностями. Ця асоціація завжди є бінарною та може існувати між двома різними сутностями або між сутністю і нею ж самою.

Основні елементи, які входять в ER-модель:

- Об'єкти предметної області (сутності);
- Безпосередньо зв'язки між об'єктами;
- Властивості об'єктів;

Моделювання зв'язків між об'єктами визначається типом зв'язку та класом належності. Основні види зв'язків:

- Один до одного;
- Один до багатьох;
- Багато до багатьох;

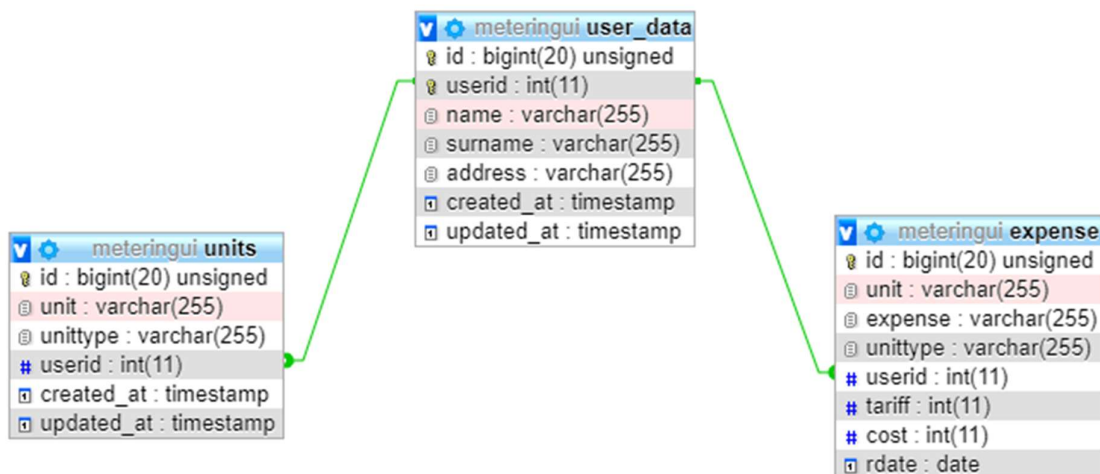


Рис. 4.1 ER-діаграма

### 4.3 Даталогічна модель системи

База даних для інформаційного ресурсу САЗПЛ складається з наступних таблиць:

Таблиця 4.1 – user\_data (Дані користувачів)

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
userid	Унікальний номер користувача в системі (зовнішній ключ)	Числовий
name	Ім'я користувача	Текстовий
surname	Прізвище користувача	Текстовий
address	Адреса користувача	Текстовий
created_at	Дата створення користувача	Дата/час
uptadet_at	Дата оновлення даних про користувача	Дата/час

Таблиця 4.2 – units (Список встановлених лічильників)

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
unit	Назва лічильника у форматі (НАЗВА-ТИП_ЛІЧИЛЬНИКА-МІСЦЕ_ВСТАНОВЛЕННЯ)	Текстовий
unittype	Тип лічильника (вода, газ, електрика)	Текстовий
userid	Унікальний номер користувача в системі (зовнішній ключ)	Текстовий
created_at	Дата створення користувача	Дата/час
uptadet_at	Дата оновлення даних про користувача	Дата/час

Таблиця 4.3 – expense (Облік витрат лічильників користувачів)

Ім'я поля	Призначення поля	Тип даних
id	Первинний ключ	Лічильник
unit	Назва лічильника у форматі (НАЗВА- ТИП_ЛІЧИЛЬНИКА- МІСЦЕ_ВСТАНОВЛЕННЯ)	Текстовий
expense	Кількість використаних одиниць продукту	Текстовий
unittype	Тип лічильника (вода, газ, електрика)	Текстовий
userid	Унікальний номер користувача в системі (зовнішній ключ)	Числовий
tariff	Ціна за одиницю продукту	Числовий
cost	Обчислена ціна за використану кількість продукту	Числовий
rdate	Дата запису показників лічильника	Дата (Y-m-d)

#### 4.4 Програми для проектування структури та контенту ресурсу

##### *PHPStorm*

JetBrains PhpStorm – комерційне інтегроване середовище розробки для PHP, яке розробляється і підтримується компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою редактор для PHP, HTML і JavaScript, з можливостями аналізу коду на ходу, відсліджування та запобігання помилок у коді і автоматизованими зобами рефакторинга для PHP, HTML і JavaScript.

Автододовнення коду в PhpStorm підтримує специфікації PHP 5.3/5.4/5.5/5.6/7.0/7.1 та новіші.

Розроблений на основі програми IntelliJ IDEA, яка в свою чергу написана на мові програмування JAVA. Користувачі мають змогу розширювати функціональність середовища розробки встановлюючи плагіни, розроблені для платформи IntelliJ, або розробити власні плагіни.

Особливості PhpStorm:

- Автодоповнення коду фіналізують класи, методи, змінні, ключові слова PHP, а також широко використовувані імена полів і змінних залежно від їхнього типу;
- Підтримка стандартів написання коду (PSR1/PSR2, Drupal, Symfony2, Zend);
- Детектор повторюваного коду;
- PHP Code Sniffer, перевіряє код на ходу;
- Підтримка редагування шаблонів Smarty (автодоповнення функцій і атрибутів Smarty, підсвічування синтаксичних помилок, автодоповнення функцій і атрибутів Smarty, автоматична вставка парних дужок, лапок тощо)

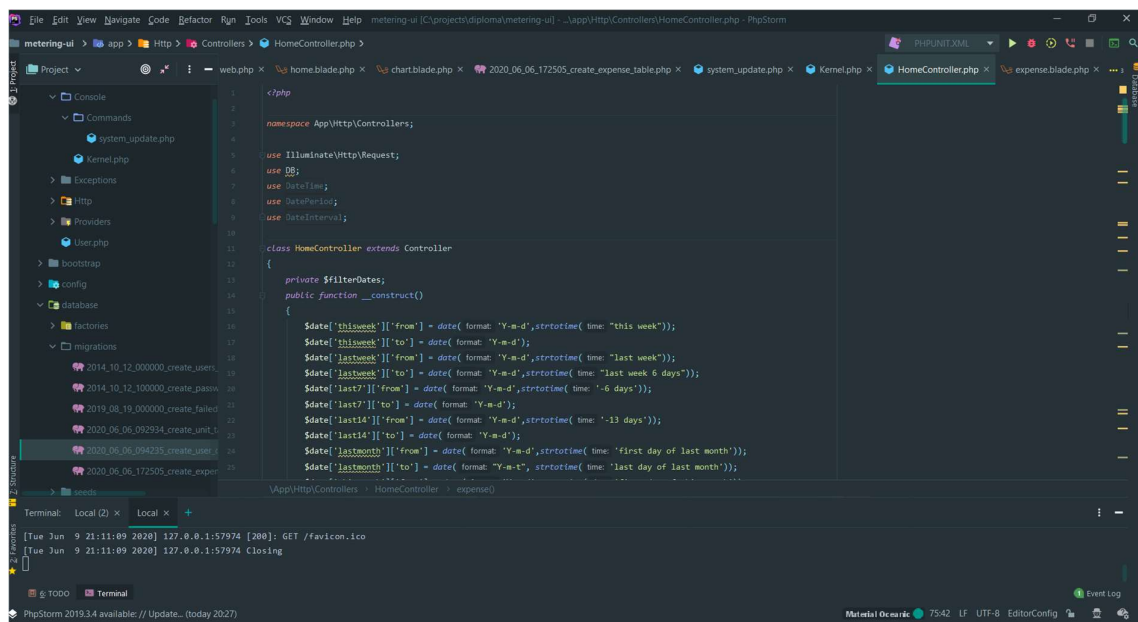


Рис. 4.2 Головне вікно PhpStorm

### *phpMyAdmin*

phpMyAdmin - це безкоштовний програмний інструмент, написаний на PHP, призначений для управління адмініструванням MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій на MySQL та MariaDB. Часто використовувані операції (керування базами даних, таблицями, стовпцями, відносинами, індексами, користувачами, дозволами тощо) можна виконувати через користувацький інтерфейс, хоча ви все ще маєте можливість безпосередньо виконувати будь-який оператор SQL.

Особливості phpMyAdmin:

- Підтримка більшості функцій MySQL:
  - переглядати та видаляти бази даних, таблиці, представлення даних, поля та індекси;
  - створювати, копіювати, видаляти, перейменувати та змінювати бази даних, таблиці, поля та індекси;
  - сервер обслуговування, бази даних та таблиці, з пропозиціями щодо конфігурації сервера;
  - виконання, редагування та закладка будь-яких SQL-висловлювань, навіть пакетних запитів;



- керувати обліковими записами та привілеями MySQL;
- керування збереженими процедурами та тригерами;
- Імпортування даних з CSV та SQL;
- Експорт даних у різні формати: CSV, SQL, XML, PDF, ISO / IEC
- Пошук у глобальному масштабі в базі даних або її підмножині

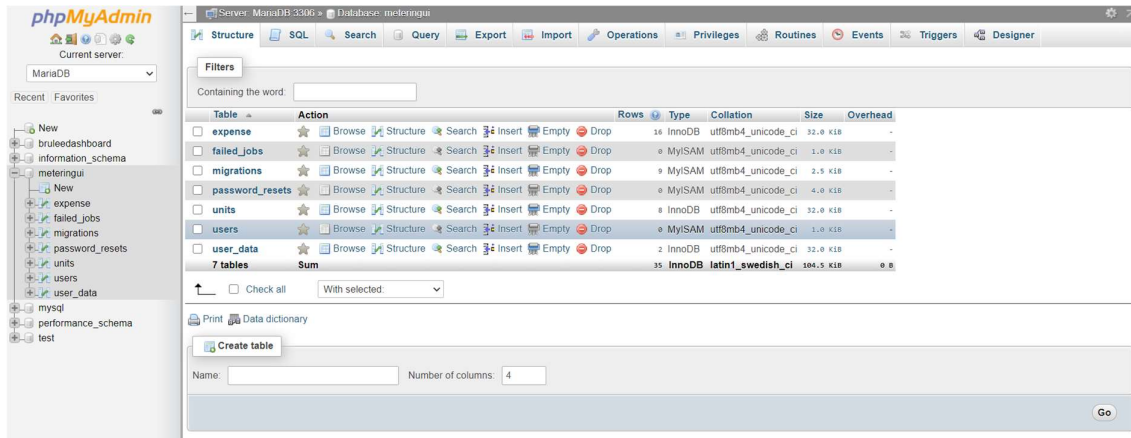


Рис. 4.3 - Головне вікно phpMyAdmin

### *WampServer*

WampServer – збірка веб-сервера, яка включає в себе Apache, MySQL, інтерпретатор PHP, phpMyAdmin та інші програми, що призначені для веб-розробки.

WampServer повністю імітує роботу віддаленого серверу, на якому в майбутньому буде розташований інформаційний ресурс САЗПЛ.

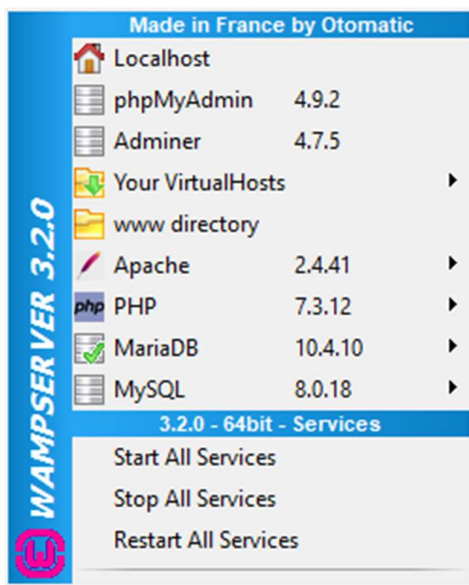


Рис. 4.4 – меню налаштувань WampServer

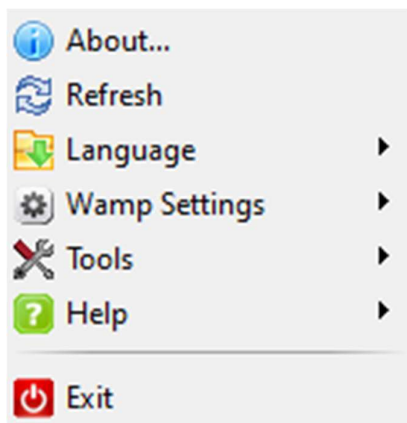


Рис. 4.5 – головне меню управління WampServer

## 4.5 Структура інформаційного ресурсу

Інформаційний ресурс призначений для обробки та відображення даних отриманих від лічильників на об'єктах користувачів.

Структура ресурсу представлена на рисунку 2.3:

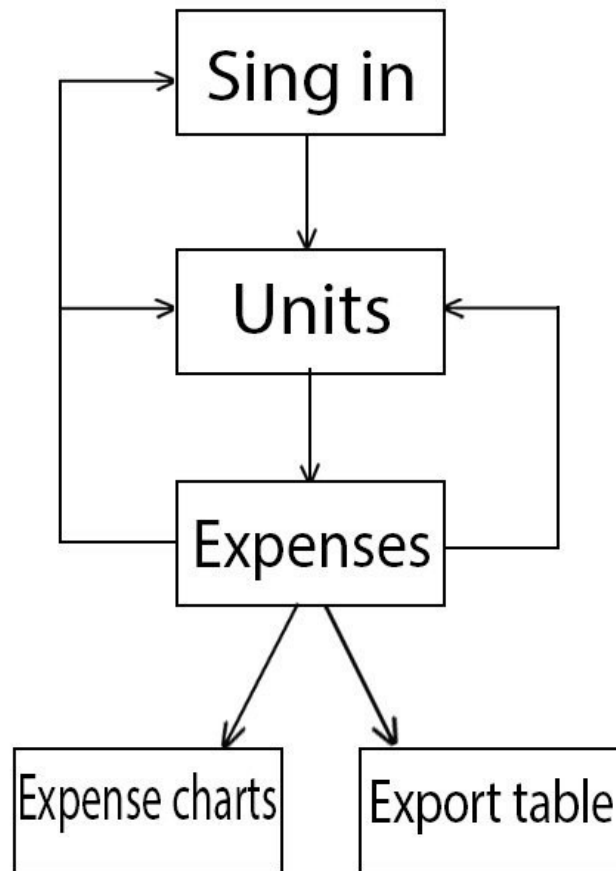


Рисунок 4.6 – структура інформаційного ресурсу САПЛЗ

#### 4.6 Висновок до розділу

Проектування бази даних для інформаційного ресурсу є найважливішим аспектом його розробки, адже база даних є фундаментом для будь-якого динамічного веб-сайту.

Якщо база даних буде занадто проста та «одновимірно» розроблена, ресурс дуже швидко почне потребувати в масштабуванні і фундаментальній переробці. Також, якщо база даних буде занадто складна, система буде програвати в швидкості конкурентам, і застосунком буде неможливо користуватись в оптимальному режимі.

Саме тому, при проектуванні бази даних для веб-застосунку САЗПЛ було враховано всі необхідні аспекти майбутньої системи і була спроектована оптимальна база даних для поставленої задачі.

## 5. Опис особливостей роботи системи

### 5.1 Встановлення фреймворку Laravel та початок роботи

Фреймворк Laravel дає змогу розгорнути пустий проект для подальшої роботи з ним. Після його встановлення, розробнику доступні всі існуючі функції та інструменти фреймворку.

Встановлення Laravel відбувається за допомогою пакетного менеджера Composer.

Composer – це інструмент для управління залежностями в PHP. Це дозволяє оголосити бібліотеки, від яких залежить ваш проект, і він буде керувати (встановлювати / оновлювати) їх для вас[11].

Для того щоб створити пустий проект-шаблон Laravel, необхідно відкрити директорію с проектом в терміналі та запустити наступні команди на виконання:

```
composer global require laravel/installer
```

За допомогою цієї команди буде встановлено інстлятор Laravel і стануть доступні команди фреймворку в терміналі.

Наступна команда створює проект з необхідними залежностями:

*laravel new metering-ui*

## 5.2 Фізична структура інформаційного ресурсу

Так як по замовчуванню фреймворк створює готову для роботи програму, розглянемо структуру проекту:

- Каталог «app» - є ядром проекту. В ньому обробляються всі функції проекту, обробляються винятки (Exceptions), створюються команди для терміналу, обробляється авторизація користувачів і т.д. ;
- Каталог «bootstrap» - містить вбудовану бібліотеку стилів ;
- Каталог «Config» - відповідає за налаштування та роботу вбудованих функцій;
- Каталог «database» - робота з базами даних за допомогою терміналу;
- Каталог «public» - загальнодоступні ресурси сайту, такі як стилі, логотип і т.д. ;
- Каталог «resources» - містить інструменти для виведення контенту на екран користувачу;
- Каталог «routes» - функції обробки URL-адреси, та перенаправлення запиту на необхідну функцію;
- Каталог «tests» - вбудовані функції тестування проекту;
- Каталог «vendor» - бібліотека функцій фреймворку що забезпечує його роботу;

Розглянемо структуру файлів що безпосередньо виконують функції роботи з даними, отриманими з лічильників.

Загальну структуру створеного функціоналу можна розділити на два пункти:

- Сутності для роботи з користувачем;
- Сутності для роботи з базами даних та системою;

До першого пункту відносяться:

- Контроллери («app/http/controllers/»)

- HomeController.php
- Controller.php
- Views (Вигляди, «resources/views/»)
  - chart.blade.php
  - expense.blade.php
  - home.blade.php
  - welcome.blade.php
- JavaScript файли («public/js/»)
  - export.js

До другого пункту відносяться:

- system\_update.php
- Kernel.php
- Файли міграцій в базу даних

### 5.3 Функції для роботи з користувачем

Так як головною задачею інформаційного ресурсу САЗПЛ є облік, зберігання та відображення даних про витрати, було реалізовано інтерфейс, за допомогою якого користувач системи має змогу знайти інформацію про витрати його системи.

Розглянемо інтерфейс користувача та його головні функції.

Для того щоб увійти в систему користувачу не потрібно створювати обліковий запис. Ключем для входу в систему є унікальний номер, який користувач отримує при встановленні САЗПЛ.

На рисунку 3.1 зображена сторінка авторизації в систему:



**Sign in**

Tax number

We'll never share your tax number with anyone else.

Continue

Рис. 5.1 – сторінка авторизації в систему

В процесі авторизації присутня перевірка на існування введеного номера користувача.

Якщо введений номер не існує, користувач отримає сповіщення про те, що такого користувача не існує.

При успішній авторизації користувача буде перенаправлено на сторінку зі списком всіх активних лічильників на його об'єкті.

Процедура авторизації реалізована в публічному методі контролера (класа) HomeController.php, home().

Введений номер надсилається на сервер в якості параметра GET запита.

Розглянемо різницю між GET та POST запитами на сервер.

GET запит створюється із рядка запиту. Все, що написано після «?» є тілом GET запиту. Наприклад:

`http://google.com?name=test&surname=test_two`

Таким чином, системний масив `$_GET` в PHP на сервері матиме вигляд

```
[  
    "name" => "test",  
    "surname" => "test_two",  
];
```

В свою чергу POST заповнюється іншим чином. Дані такого запиту зберігаються в його заголовках, тобто в прихованому від очей користувача місці.

Прийнято використовувати GET та POST наступним чином:  
GET запити використовуються якщо інформацію потрібно отримати або зчитати.

POST запити використовуються для того щоб дані відправити.



Reset user

Metering devices for Andrey Kilchitsky   Kyiv, Yangel'ya 7 <span>Expenses</span>			
#	Unit	Unit type	Start date
1	<a href="#">WM-231-wtr</a>	water	2020-06-06 00:00:00
2	<a href="#">WM-324-gas</a>	gas	2020-06-06 00:00:00
3	<a href="#">WM-159-elec</a>	electricity	2020-06-06 00:00:00
4	<a href="#">WM-485-wtr</a>	water	2020-06-06 00:00:00

Рис. 5.2 – список всіх активних лічильників на об'єкті.

Зберігання даних про користувача (повне ім'я та адреса) було створено за допомогою збереження їх в сесію за допомогою вбудованої функції фреймворку session ();

Функція session приймає як аргумент функції масив даних і зберігає їх в поточну сесію користувача. Laravel використовує файли cookie для зберігання даних про поточну сесію. Ці дані перезаписуються при кожному запуску сайту.

Користувач має змогу перейти до таблиці з обліком всіх лічильників, або, натиснувши на необхідний лічильник в таблиці, перейти до таблиці з обліком обраного лічильника.




[Reset user](#) [Back to devices list](#)

### Expenses for this month of Andrey Kilchitsky | Kyiv, Yangel'ya 7 |

[Submit filter by type](#)
[Expense Charts](#)
[Export table](#)
[Filter by date](#)

#	Unit	Unit type	Expense	Tariff	Cost	Date
			168.5		442 ₴	
1	WM-231-wtr	water	3	22 ₴	66 ₴	2020-06-07
2	WM-324-gas	gas	10	3 ₴	29 ₴	2020-06-07
3	WM-159-elec	electricity	50	2 ₴	84 ₴	2020-06-07
4	WM-485-wtr	water	2	22 ₴	44 ₴	2020-06-07
5	WM-231-wtr	water	1	22 ₴	22 ₴	2020-06-01
6	WM-324-gas	gas	12	3 ₴	35 ₴	2020-06-01
7	WM-159-elec	electricity	90	2 ₴	151 ₴	2020-06-01
8	WM-485-wtr	water	0.5	22 ₴	11 ₴	2020-06-01

Рис. 5.3 – таблиця обліку витрат

Отримання даних для таблиці обліку витрат реалізовано в публічному методі контролера (класа) `HomeController.php, expense()`.

Таблиця обліку витрат має наступні функції:

- Фільтр по необхідній даті
  - Поточний тиждень
  - Минулий тиждень
  - Останні 7 днів
  - Останні 14 днів
  - Минулий місяць
  - Поточний місяць
  - За весь час
  - Дати обрані користувачем
- Фільтр по типу лічильника
  - Газ

Зм.	Анк	№ докум.	Піппи	Лат

ДП ПМ-61 06.000.ПЗ

- Вода
- Електрика
- Графіки витрат (для кожного лічильника в поточній таблиці)
- Експорт таблиці (автоматичне завантаження поточної таблиці у форматі CSV)

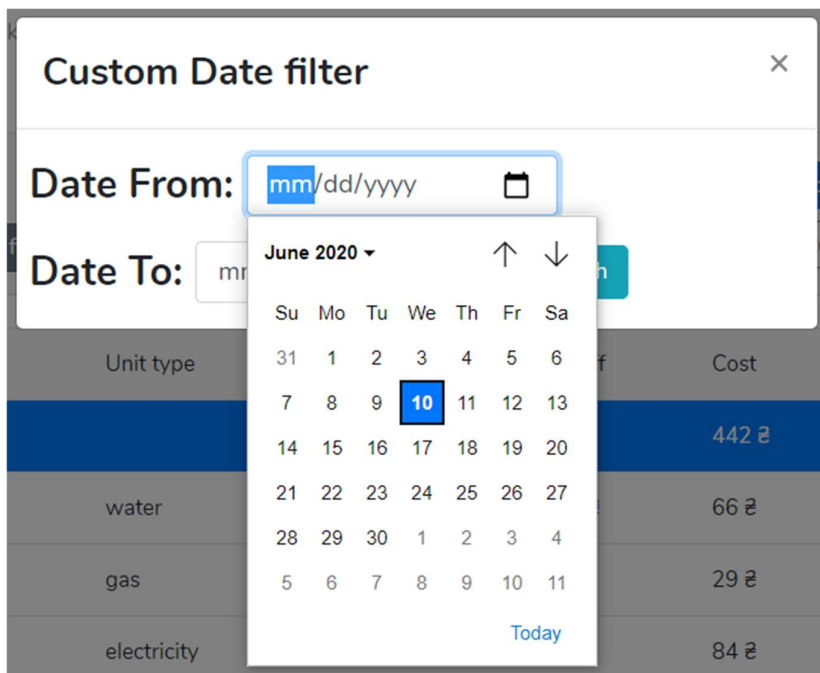


Рис 5.4 – фільтр по обраній користувачем даті

Групування та будовання фільтрів до запиту в базу даних відбувається в приватній функції `filter()` класу `HomeController`.

В якості аргументу функція приймає масив `GET`, і обробляє його в залежності від встановлених налаштувань.

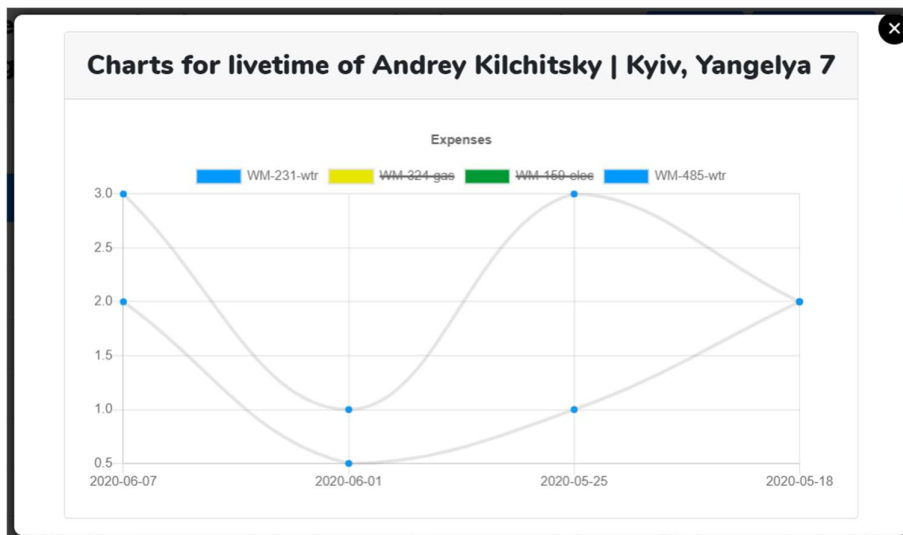


Рис 5.5 – графіки витрат лічильників

Інтерфейс графіків дає можливість приховувати та відображати потрібні лічильники. Масштабування по осі витрат відбувається автоматично.

Запит на створення графіків відправляється за допомогою AJAX функцій в JavaScript.

AJAX (Asynchronous JavaScript And XML) – підхід до побудови інтерфейсів веб-застосунків, за яких вебсторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує необхідні дані.

Рис 5.6 – фільтр по типу лічильника

На кожній сторінці ресурсу реалізовані динамічні пункти меню:

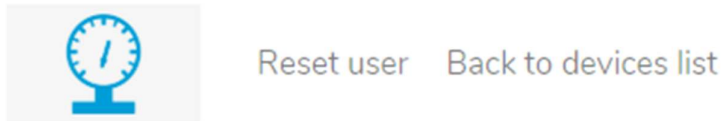


Рис 5.7 – динамічні пункти головного меню

## 5.4 Функції для роботи з базами даних

Для створення таблиць в базі даних проекту, фреймворк Laravel надає систему міграцій.

За допомогою команди:

```
php artisan make:migration CreateUnits_testTable
```

В директорії міграцій («database/migrations/») створюється файл наступної структури:

```
class CreateUnitsTestTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('units_test', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('units_test');
    }
}
```

Об'єкт `$table` відповідає за створювану таблицю. Кожен метод цього об'єкту дорівнює полю в створюваній таблиці.

Таблиці за допомогою міграцій створюються за допомогою команди:

*php artisan migrate*

Після виконання даної команди у базі даних будет створено необхідні таблиці.

Для роботи з таблицями в функціях було використано вбудований клас фреймворку «DB» та його статичні методи

- table()
- select()
- where()
- whereRaw()
- orderBy()
- groupBy()
- get()
- toArray()

### 5.5 Функції для роботи з системою

Виходячи з того, що збір і передача даних в обраній САЗПЛ буде відбуватися стаціонарним способом, необхідною особливістю системи має бути її своєчасне оновлення.

Виходячи з цього було розроблено і створено команду в терміналі для оновлення системи по розкладу.

*php artisan expenses:update*

Пристрої на об'єктах надсилають дані про витрати лічильників у форматі CSV на FTP сервер, на якому знаходиться інформаційний ресурс.

Система в автоматичному режимі запускає оновлення даних.

Також в системі встановлено поточні тарифи на одиницю витрат, тому при оновленні, обчислюється ціна за витрачену кількість одиниць, яка в свою

чергу також записується в базу даних.

Розклад оновлення буде змінюватися в залежності від кількості користувачів. При їх збільшенні, буде збільшуватись частота оновлення системи.

По замовчуванню оновлення системи відбуваються один раз на добу в 00:00.

### **5.6 Потенціал інформаційного ресурсу на розширення і вдосконалення**

На сьогоднішній день існує сильна конкуренція в галузі веб ресурсів. І головною задачею розробників є швидкий перехід від розробки до випуску застосунку в мережу.

Тому важливим аспектом є довгочасна підтримка і постійне удосконалення продукту.

Можна виділити наступні задачі на масштабування ресурсу в майбутньому:

- Програмний інтерфейс застосунку для розробників, API (надати можливість використовувати дані із САЗПЛ без використання інтерфейсу користувача)
- Розширена фільтрація даних (фільтрація по числовим значенням)
- Система авторизації для адміністраторів
- Інтеграція інтерактивної карти об'єкту користувача.

### **5.7 Висновок до розділу**

Основними принципами при розробці інтерфейсу користувача та його функцій є

- Максимальна простота в використанні та розумінні інтерфейсу;
- Реалізація необхідних користувачу функцій;

- Відловлювання всіх можливих сценаріїв поведінки користувача задля виключення будь-якої помилки сервера;
- Притримання принципу **User-friendly interface**;

При проектуванні даного застосунку перелічені принципи розробки були притримані, завдяки чому інтерфейс користувача є логічним і зрозумілим будь-якому користувачу.

## 6. Технологічний розділ

### 6.1 Опис конструкції

Прилад який є частиною системи, що розглядається в даному дипломному проекті, зображений на рис. 1 – це ультразвуковий датчик витрат.

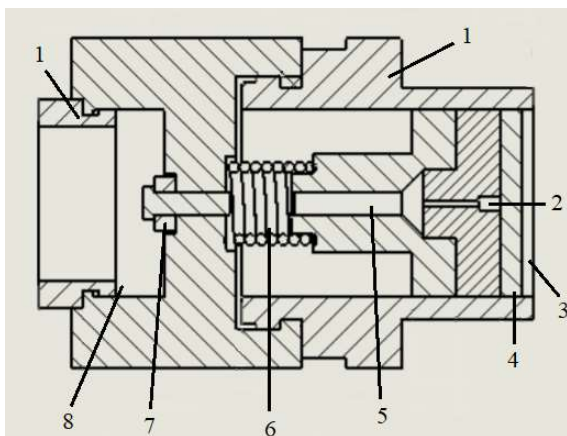


Рис 6.1 – ультразвуковий датчик витрат газу

Прилад складається з складного корпусу 1, втулки 3 і кришки 4, резонатору 2, притискача 5, пружини 6, п'єзоелемента 7 та електронного блоку 8.

Принцип дії даного приладу заснований на проходженні акустичних коливань через потік газу зв'язаних з швидкістю останнього.

### 6.2 Оцінка рівня технологічності

#### Визначення основних показників технологічності

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		5 <sup>т</sup>

Абсолютний техніко-економічний показник виготовлення  $T_v$  виражається сумою нормо-годин, витрачених на виготовлення виробу  $T_v = \sum_i T_i$ , де  $T_i$  – трудомісткість виготовлення й випробувань  $i$  – її складової частини виробу в нормо-годинах.

Трудомісткість виготовлення  $K_{ут}$  дає змогу визначити рівень технологічності конструкції відношення отриманої трудомісткості виробу  $T_v$  до початкового показника трудомісткості виготовлення  $T_{дв}$

$$K_{ут} = \frac{T_v}{T_{дв}}$$

$$K_{ут} = \frac{18}{21} = 0,85$$

Коефіцієнт уніфікації виробу  $K_y$  визначається по формулі

$$K_y = \frac{E_y + D_y}{E + D}$$

Де  $E_y$  – число уніфікованих складальних одиниць у виробі ( $E_y = 2$ );

$E_y$  – число уніфікованих деталей ( $E_y = 27$ );

$E$  – кількість складальних одиниць у приладі ( $E=3$ );

$D$  – кількість деталей, що є складовими частинами виробу ( $D=38$ );

$$K_y = \frac{2 + 27}{3 + 38} = 0,76$$

Відношення кількості уніфікованих складальних одиниць до загального числа складальних одиниць у виробі є коефіцієнтом уніфікації складальних одиниць  $K_{y.c.}$  і знаходиться за допомогою наступних маніпуляцій:

$$K_y = \frac{E_y}{E} = \frac{2}{3} = 0,69$$



Коефіцієнт уніфікації деталей виробу  $K_{у.жд.}$  знаходиться як відношення кількості уніфікованих деталей до загальної кількості деталей у виробі виключно з кріпильними.

$$K_y = \frac{D_y}{D}$$

$$K_y = \frac{27}{38} = 0,71$$

### 6.3 Розрахунок точності механоскладальних робіт

Точність показників, точність яка забезпечується технологією виготовлення деталей, аргументований вибір принципової схема, спосіб складання приладу, всі ці аспекти є важливими вимогами до конструкції. Взаємозамінність модулів, складальних одиниць та деталей – головна ідея технологічності приладу.

Припустимо, що розмір який необхідно забезпечити -  $A_{зам}$  (рис 6.2)

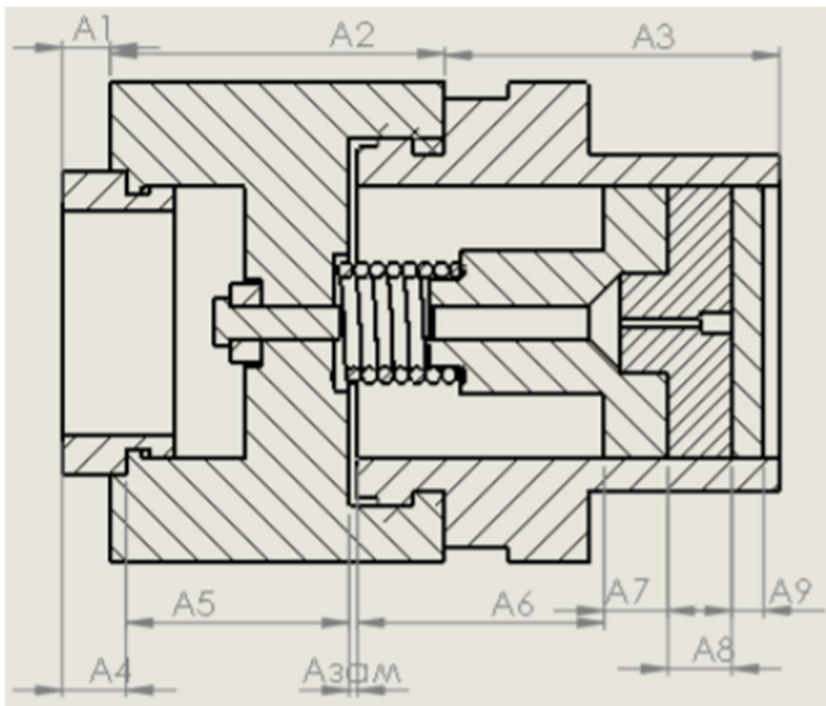


Рис. 6.2 – Датчик витрати ультразвуковий

Таблиця 6.1 – Номінальні значення ланок розмірного ланцюга та їх допустимі значення

Ланка	Номінальний Розмір, мм	Допуск, мм	Середина поля допуску, мм	Тип ланки
A1	5	$\pm 0,03$	0	<u>збільшуюча</u>
A2	20	-0,1	-0,05	<u>збільшуюча</u>
A3	20	+0,1	+0,05	<u>збільшуюча</u>
A4	4	$\pm 0,03$	0	<u>зменшуюча</u>
A5	12	+0,05	+0,025	<u>зменшуюча</u>
A6	13	+0,05	+0,025	<u>зменшуюча</u>
A7	6	$\pm 0,03$	0	<u>зменшуюча</u>
A8	6	$\pm 0,03$	0	<u>зменшуюча</u>
A9	2	$\pm 0,03$	0	<u>зменшуюча</u>

#### 6.4 Координатний метод

Знайдемо номінальне значення замкнутої ланки використавши арифметичну суму номінальних розмірів деталі. Збільшувальна ланка має додатній символ, зменшувальна ланка – від’ємний.

$$A_{\text{зам}} = (5 + 20 + 20) - (4 + 12 + 13 + 6 + 6 + 2) = 2 \text{ мм}$$

Координата середини поля допуску замикаючої ланки:

$$K_{\text{зам}} = (0,015 - 0,05 + 0,05) - (0,015 + 0,025 + 0,025 + 0 + 0 + 0) = 0,05 \text{ мм}$$

$$\delta_{\text{зам}} = 0,03 + 0,1 + 0,1 + 0,03 + 0,05 + 0,05 + 0,03 + 0,03 + 0,03 = 0,45 \text{ мм}$$

Верхнє і нижнє відхилення розміру замикаючої ланки дорівнюють

$$(BV)_{\text{зам}} = 0,05 + 0,5 * 0,45 = 0,275 \text{ мм}$$

$$(NB)_{\text{зам}} = 0,05 - 0,5 * 0,45 = 0,175 \text{ мм}$$

Таким чином, у разі повної взаємозамінності замикаюча ланка  $A_{\text{зам}}$  дорівнює:

$$A_{\text{зам}} = 2^{+0,275}_{-0,175}$$

Визначення допуску замикаючої ланки імовірнісним методом (методом неповної взаємозамінності)

$$\delta_{\text{зам}} = K \sqrt{\sum_{i=1}^{10} (\delta_i)^2}$$

Представляючи, що закони розподілу складових розмірів ланцюга є нормальними, знаходимо поле розсіювання замикаючої ланки:

$$\delta_{\text{зам}} = \sqrt{\sum_{i=1}^{10} (\delta_i)^2} = \sqrt{0,032} \approx 0,17 \text{ (мм)}$$

Визначимо координату середини поля розсіювання замикаючої ланки як:

$$\begin{aligned} \Delta K_{\text{зам}} &= \sum_{i=3}^3 (\Delta K_i + 0,5 a_i \delta_i) \\ &\quad - \sum_{i=4}^{10} (\Delta K_i + 0,5 a_i \delta_i) \\ &= (0 - 0,05 + 0,05) - (0,015 + 0,025 + 0,025) = -0,065 \text{ (мм)} \end{aligned}$$

Верхня і нижня границі поля розсіювання замикаючої ланки дорівнюють

$$(ВВ)_{\text{зам}} = \Delta K_{\text{зам}} + \frac{\delta_{\text{зам}}}{2} = 0,065 + \frac{0,17}{2} = 0,15;$$

$$(НВ)_{\text{зам}} = \Delta K_{\text{зам}} - \frac{\delta_{\text{зам}}}{2} = 0,065 - \frac{0,17}{2} = 0,02;$$

При неповній (частковій) взаємозамінності замикаюча ланка має наступний розмір:

$$A_{\text{зам}} = (1^{+0,15}_{+0,02}) \text{ мм}$$

### 6.5 Визначення фізичної (параметричної) взаємозамінності

Для вимірювання витрат природного газу в приладі застосований ультразвуковий метод. В основі цього методу лежить вимірювання часу проходження ультразвуку через вимірюване середовище. Необхідно визначити похибку часу проходження ультразвуку через газ.

Час проходження ультразвуку через середовище визначається за формулою:

$$\tau = \frac{L}{V + c * \cos(\alpha)}$$

Де L – відстань між ультразвуковими датчикам;

V – швидкість потоку;

C – швидкість ультразвуку;

$\alpha$  - кут нахилу датчиків до осі трубопроводу;

Відхилення початкових параметрів в межах допуску лічильника викликають похибку часу проходження. До таких параметрів відносяться:

$L = (82 \pm 0,5)$  мм;  $c = (439 \pm 5)$  м/с;

$\alpha = (0.61 \pm 0.01)$  рад.

Значення допусків на вказані параметри взяті з технічних умов на електродвигун.

Так як зміна параметрів в межах допусків являється випадковою, то значення похибки часу проходження ультразвуку дорівнює:

$$\tau = \sqrt{\left(\frac{\partial \tau}{\partial L} /_L \delta_L K_L\right)^2 + \left(\frac{\partial \tau}{\partial c} /_c \delta_c K_c\right)^2 + \left(\frac{\partial \tau}{\partial \alpha} /_\alpha \delta_\alpha K_\alpha\right)^2}.$$

Часткові похідні по змінних параметрах представляються як

$$\frac{\partial \tau}{\partial L} = \frac{1}{c + V * \cos(\alpha)}$$

$$\frac{\partial \tau}{\partial \alpha} = \frac{V * L * \sin(\alpha)}{(c + V * \cos(\alpha))^2}$$

$$\frac{\partial \alpha}{\partial c} = -\frac{L}{(c + V * \cos(\alpha))^2}$$

При  $V = 23$  м/с та середніх значеннях випадкових змінних коефіцієнти впливу дорівнюють

$$\frac{\partial \tau}{\partial L} = 2.18 \text{ с/мм}$$

$$\frac{\partial \alpha}{\partial c} = 0,00396 \text{ с}^2/\text{мм}$$

$$\frac{\partial \tau}{\partial \alpha} = 0,0524 \text{ мм}$$

Припускаючи, що випадкові змінні розподілені по нормальному закону, для якого  $K=1$ , отримаємо похибку часу проходження ультразвуку:

$$\begin{aligned} \delta_n &= \sqrt{(2.18)^2(0.5)^2(1)^2 + (0.00396)^2(5)^2(1)^2 + (0.00524)^2(0.01)^2(1)^2} \\ &= 21 \text{ мкс} \end{aligned}$$

Похибка часу проходження ультразвуку складає 4%. Така похибка викликана тому, що в матеріалі звукопроводу і вимірювальному середовищі не враховує заломлення звукового променя та асиметрію швидкостей.

У разі невідповідності отриманої похибки часу проходження ультразвуку вимогам експлуатації, вона зменшується шляхом більш точного регулювання відстані між ультразвуковими датчиками.

## 6.6 Розробка технологічного процесу складання виробу

Проектування технологічного процесу складання включає наступні аспекти:

- вибір раціональних процесів і способів складання;
- визначення послідовності з'єднання деталей і складальних одиниць;
- контроль виробу та складальних одиниць;
- вибір необхідного устаткування, робочих і вимірювальних інструментів;
- нормування операцій технологічного процесу.

### 6.6.1 Схема складального складу

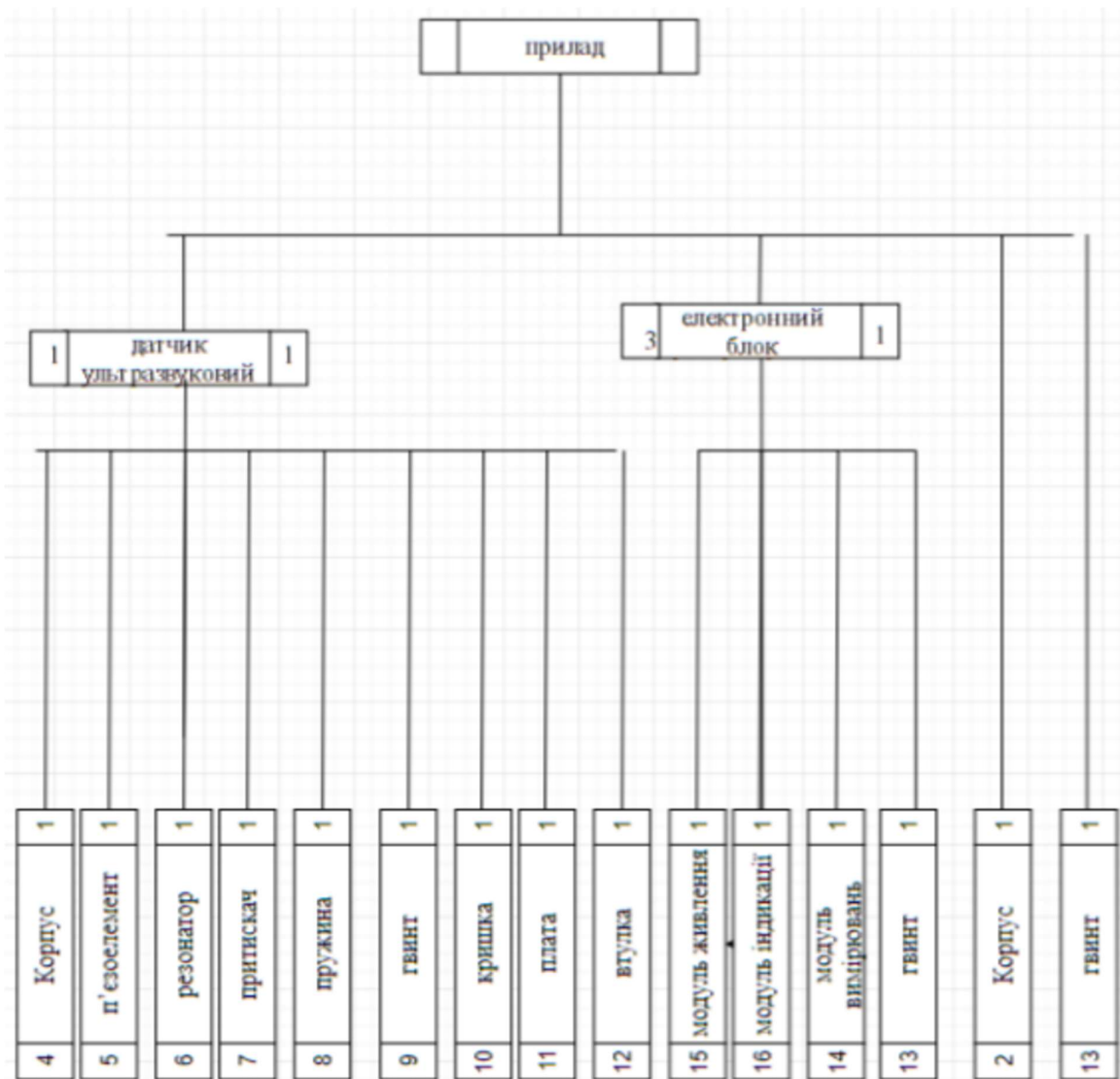


Рис. 6.3 – Схема складального складу лічильника витрати природного газу

### 6.6.2 Технологічна схема складання

Послідовність складання, способи забезпечення з'єднань, періодичність і зміст процесу регулювання, випробування та контролю визначає технологічна схема складання.

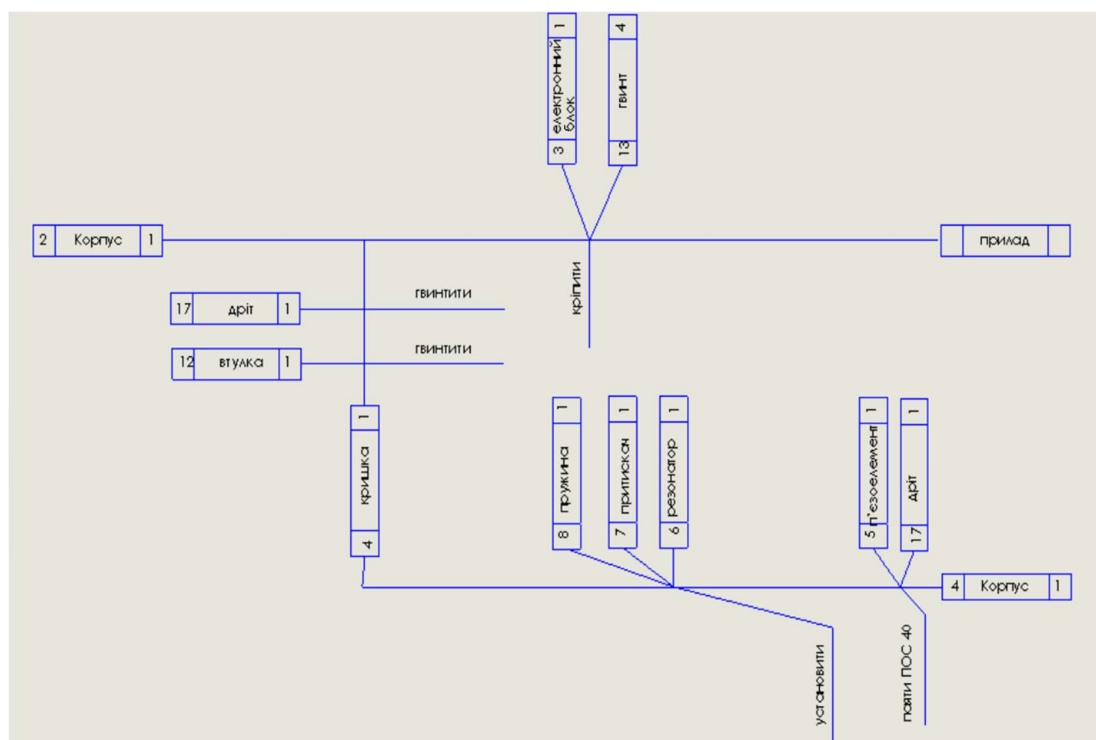


Рис. 6.4 – Технологічна карта складання.

## Висновки

1. При встановленні систем автоматичного зняття показань лічильників потрібно брати до уваги особливості будівлі в якій проходить встановлення, так як можуть виникати радіоперешкоди через особливості матеріалів в будівлі та кількості існуючих бездротових пристроїв.
2. В залежності від цілей та об'єкту на якому встановлюється система автоматичного зняття показань лічильників потрібно розглянути доцільність встановлення стаціонарної або мобільної систем зчитування.
3. Так як система автоматичного зняття показань лічильників на пряму залежить від передачі даних через інтернет, найважливішу роль грає стабільний і безперебійний зв'язок с мережею інтернет. Тому питання якості зв'язку з мережею інтернет впливає на вибір способу зчитування даних.
4. Виходячи з того, що всі дані передаються на веб ресурс, необхідно забезпечити швидкий і легкий доступ до нього. Саме тому було обрано даний спосіб, адже для отримання даних потрібен тільки браузер, котрий встановлений по замовчуванню на всіх комп'ютерах та на 99% мобільних телефонів.
5. Опираючись на обрану систему автоматичного зчитування показників лічильників, було опрацьовано існуючі аналоги систем та обрано оптимальні складові цієї системи. Обрані складові системи відповідають стаціонарному способу зчитування даних. Проаналізовані прилади мають підтримку необхідних функцій і особливостей системи.
6. Було розроблену просту у розумінні і структурі базу даних, яка в свою чергу відповідає всім поставленим вимогам системи.



7. Так як головною задачею при розробці інформаційного ресурсу була простота у використанні та реалізація всіх основних функцій обліку даних, було розроблено застосунок, який повністю відповідає поставленим вимогам, має інтуїтивний інтерфейс та швидку роботу з базами даних.
8. Була проведена оптимізація запитів до баз даних та зменшений час відгуку сервера. Швидкість роботи є одним із найважливіших аспектів роботи будь-якого застосунку.

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		6 <sup>5</sup>

## Список літератури

1. PHP для начинающих. — СПб.: Питер, 2014. — 288 с.: ил. ISBN 978-5-496-00844-0.
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста») ISBN 5-272-00355-1
3. JavaScript [Електронна адреса]  
[https://developer.mozilla.org/ru/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/JavaScript_basics)
4. SQL - ПРЕИМУЩЕСТВА И НЕДОСТАТКИ [Електронна адреса]  
<https://www.softbusiness.net/Default.aspx?id=29>
5. Ополева Г.Н. Схемы и подстанции электроснабжения. — М.: Инфра-М, 2006. — 481 с.
6. Цупрун А. Ю. Системы управления процессами и механизмами машин непрерывного литья заготовок. Киев: Украинская Ассоциация Сталеплавильщиков;
7. JavaScript [Електронна адреса]: Вікіпедія. Вільна енциклопедія. —Режим доступу : <https://uk.wikipedia.org/wiki/JavaScript>;
8. SQL [Електронна адреса]: Вікіпедія. Вільна енциклопедія. —Режим доступу : <https://uk.wikipedia.org/wiki/SQL>.
9. Laravel [Електронна адреса]: Вікіпедія. Вільна енциклопедія. —Режим доступу : <https://uk.wikipedia.org/wiki/Laravel>.
10. Автоматизована система збору даних з приладів обліку (вимірювання) енергоресурсів (АСЗД) для побутових споживачів [Електронна адреса]: <http://aim-ltd.kiev.ua/ua/K/id/180713-15764>
11. Composer [Електронна адреса]: <https://getcomposer.org/doc/00-intro.md>
12. PHP [Електронна адреса]:  
<https://sites.google.com/site/da2217lw4/home/php>

					ДП ПМ-61 06.000.ПЗ	
Зм.	Анк	№ докум.	Піппи	Лат		66

13.phpMyAdmin [Електронна адреса]: <https://www.phpmyadmin.net/>

					ДП ПМ-61 06.000.ПЗ	
Зм.	Арк	№ докум.	Пілпи	Лат		67